

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

*Факультет інформатики та обчислювальної техніки*

(повне найменування інституту, факультету)

*Автоматизованих систем обробки інформації і управління*

(повна назва кафедри)

«До захисту допущено»

**В.о. завідувача кафедри**

\_\_\_\_\_ О.А.Павлов  
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**Дипломний проект**

**на здобуття ступеня бакалавра**

з напрямку підготовки \_\_\_\_\_ **6.050103 «Програмна інженерія»**  
на тему \_\_\_\_\_ **Web-застосування для пошуку роботи**

**Виконав:** студент IV курсу, групи ІІІ-51 Миколаєнко Максим Олександрович  
(прізвище, ім'я, по батькові) \_\_\_\_\_ (підпис)

**Керівник** \_\_\_\_\_ ст. викл. к.ф-м.н. Клименко О.М  
\_\_\_\_\_ (підпис)  
посада, науковий ступінь, вчене звання, прізвище, ініціали

**Консультант  
з графічної  
документації** \_\_\_\_\_ доц., к.т.н., Ліщук К.І  
\_\_\_\_\_ (підпис)  
посада, науковий ступінь, вчене звання, прізвище, ініціали

**Рецензент:** \_\_\_\_\_ проф. каф. ММСА ІПСА, д.т.н., Бідюк П.І.  
\_\_\_\_\_ (підпис)  
посада, науковий ступінь, вчене звання, прізвище, ініціали

Засвідчую, що у цьому дипломному проекті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) Інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) – 6.050103  
«Програмна інженерія» (Програмне забезпечення систем)

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

О.А. Павлов  
(підпис) (ініціали, прізвище)

“ ” 2019 р.

**ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Миколаєнко Максиму Олександровичу  
(прізвище, ім'я, по батькові)

**1. Тема проекту « Web-застосування для пошуку роботи»**

керівник проекту Клименко Олена Миколаївні, к.ф-м.н., старший викладач  
( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “23” квітня 2019 р. №1181-с

**2. Термін подання студентом проекту «03» червня 2019 року**

**3. Вихідні дані до проекту**

*Технічне завдання*

**4. Зміст пояснювальної записки**

*1) Аналіз вимог до програмного забезпечення: основні положення,*

*опис предметної області, огляд існуючих технічних рішень та відомих*

*програмних продуктів, розробка функціональних та нефункціональних вимог,*

*постановка завдань*

*2) Моделювання, архітектура та конструювання програмного забезпечення:*

*аналіз програмного забезпечення, вхідні та вихідні дані, засоби розробки,*

*структура бази даних та класів*

*3) Аналіз якості та тестування програмного забезпечення*

4) Розгортання та впровадження програмного забезпечення

## 5. Перелік графічного матеріалу

1) Схема структурна варіантів використань

2) Схема бази даних

3) Схема структурна класів програмного забезпечення

## 6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «12» березня 2019 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення рекомендованої літератури		
2.	Аналіз існуючих методів розв'язання задачі		
3.	Постановка та формалізація задачі		
4.	Аналіз вимог до програмного забезпечення		
5.	Алгоритмізація задачі		
6.	Моделювання програмного забезпечення		
7.	Обґрунтування використовуваних технічних засобів		
8.	Розробка архітектури програмного забезпечення		
9.	Розробка програмного забезпечення		
10.	Налагодження програми		
11.	Виконання графічних документів		
12.	Оформлення пояснювальної записки		
13.	Подання ДП на попередній захист		
14.	Подання ДП рецензенту		
15.	Подання ДП на основний захист		

Студент \_\_\_\_\_ Миколаєнко М.О.  
(підпис)

Керівник проекту \_\_\_\_\_ Клименко О.М.  
(підпис)

[illegible]

# **Пояснювальна записка до дипломного проекту**

на тему: Web-застосування для пошуку роботи

Київ – 2019 року



### АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з шести розділів, містить 11 рисунків, 92 таблиці, 10 джерел.

Дипломний проект призначений для створення резюме та пошуку роботи відповідно до даних, що вказані у створеному резюме. Для цього в дипломному проекті створено веб-додаток, який дозволяє вирішувати дані задачі.

У першому розділі було здійснено опис та аналіз предметної області розробки. Описано та проаналізовано декілька програмних продуктів подібних до дипломного проекту. Визначено призначення розробки даного застосування. Наведені цілі та задачі розробки.

У другому розділі проведено опис бізнес-процесів за допомогою діаграм BPMN, визначення архітектуру веб-додатку, обрано технології, за якими буде розроблятися дипломний проект.

У третьому розділі проаналізовано якість програмного забезпечення, описані процеси тестування, та наведено три контрольних приклади тестування.

У четвертому розділі було описано процес впровадження та супроводу сервісу, зокрема розгортання програмного забезпечення та створено інструкцію користувача та адміністратора.

ПОШУК РОБОТИ, РЕЗЮМЕ, ВАКАНСІЇ, ВЕБ-ДОДАТОК

					КП.ІП-5114.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

**ABSTRACT**

Structure and scope of work. The explanatory note of the diploma project consists of six chapters, containing 11 drawings, 92 tables, and 10 sources.

The graduate work is designed to solve such tasks as creating a resume and job search using the specific data in the created resume. The web application has been created to solve the aim.

In the first section, the description and analysis of the subject area of development was made. Several software products similar to the diploma project are described and analyzed. The purpose of the development of this application is determined. The presented goals and tasks of development.

The second section describes business processes using BPMN charts, defines the architecture of the web application, and selects the technologies for which a graduation project will be developed.

The third section analyzes the quality of the software, describes the testing processes, and provides three benchmarking tests.

The fourth section described the implementation and maintenance of the service, including the deployment of the software, and created a user and administrator's instructions.

**SEARCH OF WORK, RESUME, VACANCIES, WEB APPLICATION**

					КПІ.ІП-5114.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6



КПІ.ІП-5114.045440.01.81

					КПІ.ІП-5114.045440.01.81	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....</b>	<b>10</b>
<b>ВСТУП.....</b>	<b>11</b>
<b>1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>12</b>
1.1 ЗАГАЛЬНІ ПОЛОЖЕННЯ .....	12
1.2 ЗМІСТОВНИЙ ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	12
1.2.1 Резюме.....	12
1.2.2 Вакансія.....	14
1.3 АНАЛІЗ УСПІШНИХ ІТ-ПРОЕКТІВ.....	14
1.3.1 Аналіз відомих програмних продуктів.....	15
1.4 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	16
1.4.1 Розроблення функціональних вимог.....	17
1.4.2 Розроблення нефункціональних вимог .....	58
1.4.3 Постановка комплексу завдань модулю .....	58
1.5 Висновки по розділу .....	59
<b>2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>60</b>
2.1 МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	60
2.2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	68
2.3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	70
2.4 АНАЛІЗ БЕЗПЕКИ ДАНИХ.....	86
2.5 Висновки до розділу .....	87
<b>3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>88</b>
3.1 АНАЛІЗ ЯКОСТІ ПЗ.....	88
3.2 ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ.....	89
3.3 ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ .....	90
3.4 Висновки до розділу .....	91
<b>4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ....</b>	<b>92</b>
4.1 РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	92
4.1.1 Перенесення проекту.....	92

4.1.2	Перенесення даних з бази даних.....	92
4.2	РОБОТА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	92
4.3	Висновки до розділу .....	92
<b>ВИСНОВКИ .....</b>		<b>93</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ.....</b>		<b>93</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

**HR** - Human Resources - особа, яка займається підбором кадрів.

**BPMN** - Business Process Model and Notation - система умовних позначень (нотація) для моделювання бізнес-процесів.

**HTML** - HyperText Markup Language - мова розмітки для створення веб-сторінок і веб-додатків.

**JQuery** - популярна JavaScript-бібліотека з відкритим сирцевим кодом.

**AJAX** - Asynchronous Javascript and XML – підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані.

**MySQL** - це одна з найпопулярніших і найпоширеніших СУБД (система управління базами даних) в інтернеті.

**PHP** - Hypertext Preprocessor - скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера, є однією з найпоширеніших мов програмування, що використовуються у сфері веб-розробок.

**API** - Application Programming Interface - це інтерфейс прикладного програмування, який представляє собою набір готових класів, процедур, функцій, структур і констант, які надаються сервісом для використання у зовнішніх програмних продуктах.

**JSON** - JavaScript Object Notation - це текстовий формат обміну даними між комп'ютерами. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу.

**BCrypt** - адаптивна криптографічна функція формування ключа, що використовується для безпечного зберігання паролів.

## ВСТУП

В наш час студенту важко знайти роботу після закінчення університету, якщо він її не знайшов ще під час навчання у ВНЗ. Оскільки роботодавцям потрібні працівники вже з досвідом роботи, бо вони не хочуть витратити гроші і час на навчання недосвідчених працівників. Звісно, якщо це не компанії, які шукають собі людей на стажування, або студент, що випустився не володіє усіма навичками, що потрібні, і без досвіду роботи. Через це студенти часто після невдалих спроб пошуку роботи за спеціальністю йдуть працювати не за своїм фахом. Це виникає з багатьох причин, однією з яких є невдало складене резюме, або взагалі його відсутність, та неправильна стратегія пошуку роботи.

Для того, щоб студенту знайти роботу, потрібно, щоб компанії про нього дізналися, а для цього йому потрібно скласти резюме. Звісно зараз є багато веб-сервісів, які дозволяють знайти роботу, а також скласти резюме, але немає єдиного сервісу, який допоможе не просто створювати резюме, в правильному вигляді, а шукати вакансії одразу з декількох сайтів-джерел для пошуку роботи, які підходять для того, щоб відправити туди щойно створене резюме. Оскільки такого веб-додатку немає або, якщо є, то він платний, виникла ідея зробити такий.

Планується, що такий веб-додаток зможе не тільки допомогти студентам у пошуку роботи своєї мрії, але й отримати інформацію щодо популярності тої чи іншої мови програмування у даний час.

Також в дипломному проекті є можливість для роботодавців створювати свої вакансії, які також будуть пропонуватися шукачам роботи після створеного резюме.

Дипломний проект ставить за мету створення веб-сервісу для створення резюме та автоматизацію пошуку вакансій, які підходять створеному резюме.

Планується, що веб-додаток буде корисний насамперед студентам, які активно шукають роботу за фахом.

					КП.ІП-5114.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

# 1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 1.1 Загальні положення

Пошук роботи через мережу Інтернет – найпопулярніший та доступний метод пошуку роботи. Тому що в інтернеті спілкування роботодавців із потенційними працівниками є найдемократичнішим. Тут кожен може обрати для себе найбільш оптимальний час та місце роботи, а також пристойну (на його думку) винагороду за власну діяльність.

Сайти з пошуку роботи бувають як загальними, так і спеціалізованими. На деяких таких сайтах інформація про роботодавців є загальнодоступною. Однак на більшості сайтів такого напрямку даної інформації у відкритому вигляді немає. Тому шукачі роботи можуть відгукнутися на цікаві їм вакансії лише за допомогою розміщеного на сайті резюме.

Серед переваг пошуку роботи в мережі Інтернет є те, що на сайтах з працевлаштування великий вибір. Створені умови для швидкого пошуку необхідної інформації. За кількістю вакансій Інтернет значно перекидає паперову пресу.

Серед недоліків - потрібно мати підключення до мережі інтернет. Вакансії, які відображаються на сайтах залежать від політики модерації цього сайту. На ненадійних сайтах можлива наявність неправдивих вакансій від недобросовісних роботодавців, або просто від агенцій, які збирають бази резюме. Велика кількість людей має доступ до інформації та користується нею.  
[1]

## 1.2 Змістовний опис і аналіз предметної області

### 1.2.1 Резюме

Резюме відіграє важливу роль при працевлаштуванні. Це зумовлено тим, що в наш час резюме стало невід'ємною частиною працевлаштування. Такий підхід набуває все більшого розповсюдження, тому, що характерною рисою

					КП.ІП-5114.045440.01.81	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

функціонування ринку праці є гостра конкуренція між претендентами на кожне вакантне місце. У конкурентній боротьбі перемагають не тільки більш кваліфіковані та добросовісні працівники, а й ті, які зуміли добре представити свої переваги як майбутнього члена певної фірми (організації, установи) на етапі відбору претендентів на робоче місце.

Резюме являє собою документ, в якому претендент на вакантне місце подає роботодавцю стислу позитивну інформацію про себе. Воно служить підставою для запрошення на співбесіду з роботодавцем. Слід підкреслити, що резюме найперший документ, за яким роботодавець складає портрет претендента на вакантну позицію. По ньому формується перше і, як відомо, найбільш важливе враження про людину. Резюме являє собою продуманий і ретельно складений документ, який має справити миттєве позитивне враження на потенційного роботодавця.

Треба усвідомити, що резюме не статичний, а динамічний документ, який потребує постійного оновлення.

Резюме має бути коротким (до двох сторінок). Разом з тим роботодавець повинен мати досить повне уявлення про професійний багаж претендента на вакантне місце. Резюме – це документ, який висвітлює головні досягнення у трудовій діяльності, вміння та попередній досвід роботи.

Структура та основні вимоги до складання резюме:

- а) прізвище, ім'я, по батькові;
- б) основні особисті дані;
- в) кваліфікація;
- г) досвід роботи;
- д) освіта;
- е) знання мов;
- є) додаткова інформація.[2]

### 1.2.2 Вакансія

Вакансія - це вільне робоче місце в штаті будь-якого підприємства, на яке може претендувати будь-яка людина, яка має бажання і потрібні навички.

Простими словами можна сказати, що вакансія є робоче місце, яке може бути зайняте працівником, який має відповідну кваліфікацію та навички.

Вакансії на сайті роботодавця можуть бути відкритими, закритими і анонімними:

- відкрита або пряма вакансія передбачає вільний її розгляд, відразу ж після її оприлюднення або публікації на сайті підприємства;
- закрита вакансія ніде не оголошується, проте підприємство її резервує на випадок, коли особа, яка добивається роботу буде цікавою роботодавцю, як фахівець. Таку вакансію можна зайняти тільки в тому випадку, якщо Ви самі будете пропонувати себе, як висококваліфікованого працівника;
- анонімна вакансія не передбачає прямої вказівки на посаду, на підприємство, яке її надає, а позначається в розмитому вигляді типу «фінансової компанії потрібні співробітники». До таких вакансіях потрібно ставитись з обережністю, адже не зрозуміло навіщо підприємству приховувати себе.[3]

### 1.3 Аналіз успішних ІТ-проектів

На сьогоднішній день все більше сфер життєдіяльності переходить в онлайн, і пошук роботи не став виключенням. Зараз вже важко зустріти людину, яка шукає роботу через оголошення в газеті чи в спеціалізованому журналі, тому що вони також вже майже повністю перейшли в онлайн.

Пошук роботи в онлайн це швидко та зручно, адже не треба гортати сторінки в пошуках потрібної сторінки, на якій знаходяться потрібні тобі вакансії, а потім вичитувати все, щоб знайти саме ті вакансії, які дійсно підходять по рівню твоїх знань в тому чи іншому напрямках, чи шукати, де ти



залишив цю газету чи журнал, або слідкувати за тим, щоб з нею нічого не трапилось.

В інтернеті все набагато простіше, просто заходиш на один з багатьох сайтів з пошуку роботи, пишеш назву вакансії, яку шукаєш, вибираєш в фільтрах необхідні налаштування, і тобі відображають список вакансій, які повністю підходять за тими критеріями, які ти виставив у фільтрах. Або є ще варіант зайти на сайт, створити резюме, залишити його відкритим для перегляду, та чекати, доки тобі не запропонують роботу HR з компаній, які шукають на вакантні місця спеціалістів твого профілю. І вже не ти будеш шукати роботу та витрачати свій час, а тебе будуть шукати люди, які займаються цим на своїй посаді.

### 1.3.1 Аналіз відомих програмних продуктів

Найбільш популярними представниками пошуку роботи в інтернеті є сайти Work.ua, Rabota.ua, HeadHunter Україна, Dou.ua, OLX.

Знизу приведу опис деяких з них:

#### а) Work.ua :

- наявність розділу «Вакансії для переселенців»;
- фахівці всіх рівнів, можливість створити резюме за допомогою конструктора;
- можливість прикріпити і відправити резюме роботодавцю без реєстрації;
- найбільший в Україні сайт відповідної тематики, зараз тут доступно понад 80 тисяч вакансій.

#### б) Rabota.ua :

- можливість створити резюме (до 5-ти) за допомогою конструктора;
- можливість прикріпити і відправити резюме роботодавцю без реєстрації;

- другий за популярністю сайт, безліч вакансій, окрема можливість знайти першу роботу.

в) HeadHunter Україна (hh.ua) :

- цільова аудиторія – топ-менеджери і фахівці вузько кваліфікованих напрямків;

- позиціонує себе, як міжнародний сайт з пошуку роботи та персоналу, представлено понад 54 тисячі компаній та близько 1 млн резюме.[4]

Таблиця 1.1 – Порівняння відомих наявних продуктів

	Список статей за темою пошуку	Список курсів для навчання	Орієнтовано на студентів	Популярна платформа	Велика кількість вакансій та резюме	Багатомовний
Work.ua	+	-	-	+	+	+
Rabota.ua	-	-	-	+	+	+
hh.ua	-	-	-	+	+	+
Диплом	+	+	+	-	-	-

#### 1.4 Аналіз

Щоб досягти мети дипломного проекту, до нього висувається низка вимог, які повинні забезпечити роботу користувачів з системою. Користувачі системи діляться на 3 типи:

- пошукач;
- роботодавець;
- адміністратор.

## 1.4.1 Розроблення функціональних вимог

Дипломний проект може мати наступні варіанти використання:

Таблиця 1.2 – Варіант використання UC001

Назва	Авторизація користувача.
Опис	Користувач має можливість авторизуватись в системі.
Учасники	Користувач.
Передумови	Користувач вже зареєстрований в системі.
Постумови	Користувач перенаправляється до особистого кабінету.
Основний сценарій	1) система відображає вікно авторизації, яке містить логін та пароль; 2) користувач заповнює поля вводу; 3) користувач натискає кнопку “Вхід”; 4) система авторизує користувача;
Розширення сценаріїв	4.1) система виявляє, що користувач ввів некоректні данні; 4.1.а) система демонструє повідомлення на екрані, яку саме помилку допустив користувач.

Таблиця 1.3 – Варіант використання UC002

Назва	Реєстрація користувача.
Опис	Користувач може зареєструватись в системі.
Учасники	Користувач.
Передумови	Користувач ще не зареєстрований в системі.

Продовження таблиці 1.3

Постумови	Користувач зареєстрований в системі.
Основний сценарій	<p>1) система відображає вікно реєстрації з полями «Логін», «Пароль», «Повторіть пароль», можливістю обрати в якості кого реєструється користувач «Зареєструватись як» та кнопку «Реєстрація»;</p> <p>2) користувач заповнює всі поля, та вибирає в якості кого він реєструється;</p> <p>3) натискає на кнопку «Реєстрація» та переходить на сторінку авторизації;</p>
Розширення сценаріїв	<p>3.1) система перевіряє чи існує ще в когось з користувачів даний логін;</p> <p>3.1.а) система повідомляє користувача про існування такого логіна;</p> <p>3.2) система перевіряє чи введений пароль в полі «Повторіть пароль» співпадає з паролем в полі «Пароль»;</p> <p>3.2.а) система повідомляє користувача що паролі не співпадають.</p>

Таблиця 1.4 – Варіант використання UC003

Назва	Створення резюме.
Опис	Користувач з типом «Пошукач» має можливість створити резюме.
Учасники	Користувач.
Передумови	Користувач авторизований в системі.

## Продовження таблиці 1.4

Постумови	Резюме створено.
Основний сценарій	<p>1) користувач переходить на сторінку створення резюме; користувач заповнює обов'язкові поля «Ім'я», «Прізвище», «Дата народження», «Місто», «Номер телефону», «E-mail», «Бажана посада», «Бажана заробітна плата», «Вид зайнятості», «Рубрика резюме», «Навички», та необов'язкові, інформацію про досвід роботи в полях «Компанія», «Сфера діяльності компанії», «Посада», «Початок роботи», «Кінець роботи», «Обов'язки та досягнення», інформацію про освіту в полях «Рівень освіти», «Назва ВНЗ», «Факультет», «Спеціальність», «Рік початку», «Рік закінчення», інформацію про володіння мовами в полях «Мова», «Рівень володіння», та додаткова інформація в полі «Додаткова інформація про себе»;</p> <p>2) користувач натискає на кнопку “Зберегти”;</p> <p>3) система створює нове резюме;</p>
Розширення сценаріїв	<p>3.1) система перевіряє чи заповнені обов'язкові поля;</p> <p>3.2) система повідомляє користувача про необхідність заповнити обов'язкові поля.</p>

## Таблиця 1.5 – Варіант використання UC004

Назва	Редагування резюме.
Опис	Користувач з типом «Пошукач» має можливість редагувати свої резюме.

Продовження таблиці 1.5

Учасники	Користувач.
Передумови	Користувач авторизований в системі.
Постумови	Резюме відредаговано.
Основний сценарій	1) користувач переходить на сторінку редагування резюме; 2) користувач редагує потрібні йому поля; 3) користувач натискає на кнопку “Зберегти”; 4) система зберігає зміни в резюме;
Розширення сценаріїв	4.1) система перевіряє чи заповнені обов’язкові поля; 4.2) система повідомляє користувача про необхідність заповнити обов’язкові поля.

Таблиця 1.6 – Варіант використання UC005

Назва	Видалення створеного резюме.
Опис	Користувач з типом «Пошукач» має можливість видалити свої резюме.
Учасники	Користувач.
Передумови	Користувач авторизований в системі.
Постумови	Обране резюме видалене.
Основний сценарій	1) користувач обирає резюме, яке хоче видалити; 2) користувач натискає кнопку “Видалити”; 3) система видаляє резюме.

## Продовження таблиці 1.6

Розширення сценаріїв	
----------------------	--

Таблиця 1.7 – Варіант використання UC006

Назва	Перегляд резюме.
Опис	Користувач з типом «Пошукач» має можливість переглядати свої резюме.
Учасники	Користувач.
Передумови	Користувач авторизований в системі.
Постумови	Надано список створених користувачем резюме.
Основний сценарій	1) користувач відкриває список створених резюме; 2) система відображає список створених користувачем резюме.
Розширення сценаріїв	

Таблиця 1.8 – Варіант використання UC007

Назва	Пошук курсів.
Опис	Користувач з типом «Пошукач» має можливість знайти курси по вибраній мові програмування.
Учасники	Користувач.
Передумови	Користувач авторизований в системі.

Продовження таблиці 1.8

Постумови	Надано список курсів за вибраною мовою програмування.
Основний сценарій	1) користувач переходить на сторінку пошуку курсів; 2) користувач вибирає по якій мові програмування він хоче знайти курси; 3) користувач натискає кнопку «Знайти»; 4) система відображає список курсів, де користувач може отримати навички по вибраній мові програмування.
Розширення сценаріїв	

Таблиця 1.9 – Варіант використання UC008

Назва	Перегляд статистики вакансій з сайтів пошуку роботи.
Опис	Користувач має можливість переглянути статистику вакансій з сайтів пошуку роботи.
Учасники	Користувач.
Передумови	Користувач авторизований в системі.
Постумови	Відображено кругову діаграму з порівняльною статистикою.
Основний сценарій	1) користувач переходить на сторінку статистики; 2) користувач вибирає по яким критеріям він хоче отримати статистику вакансій; 3) користувач натискає кнопку «Показати статистику» ; 4) система відображає кругову діаграму з порівняльною статистикою.



## Продовження таблиці 1.9

Розширення сценаріїв	
----------------------	--

Таблиця 1.10 – Варіант використання UC009

Назва	Оновлення статистики вакансій з сайтів пошуку роботи.
Опис	Користувач має можливість оновити статистику вакансій з сайтів пошуку роботи.
Учасники	Користувач.
Передумови	Користувач авторизований в системі.
Постумови	Статистику оновлено.
Основний сценарій	1) користувач переходить на сторінку статистики; 2) користувач натискає кнопку «Оновити статистику»; 3) система оновлює статистику і повідомляє про це користувача.
Розширення сценаріїв	

Таблиця 1.11 – Варіант використання UC010

Назва	Створення вакансії.
Опис	Користувач з типом «Роботодавець» має можливість створити вакансію.
Учасники	Користувач.

Продовження таблиці 1.11

Передумови	Користувач авторизований в системі.
Постумови	Вакансію створено.
Основний сценарій	<p>1) користувач переходить на сторінку створення вакансії; користувач заповнює обов'язкові поля «Ваше ім'я», «Компанія», «Місто», «Номер телефону», «E-mail», «Адреса», «Посада», «Заробітна плата», «Вид зайнятості», «Рубрика», «Навички», «Рівень освіти», «Мова», «Рівень володіння», та додаткова інформація в полі «Детальний опис вакансії» ;</p> <p>2) користувач натискає на кнопку “Зберегти”;</p> <p>3) система створює нову вакансію;</p>
Розширення сценаріїв	<p>3.1) система перевіряє чи заповнені обов'язкові поля;</p> <p>3.2) система повідомляє користувача про необхідність заповнити обов'язкові поля.</p>

Таблиця 1.12 – Варіант використання UC011

Назва	Редагування вакансії.
Опис	Користувач з типом «Роботодавець» має можливість редагувати свої вакансії.
Учасники	Користувач.
Передумови	Користувач авторизований в системі.
Постумови	Вакансію відредаговано.

## Продовження таблиці 1.12

Основний сценарій	1) користувач переходить на сторінку редагування вакансії; 2) користувач редагує потрібні йому поля; 3) користувач натискає на кнопку “Зберегти”; 4) система зберігає зміни у вакансії;
Розширення сценаріїв	4.1) система перевіряє чи заповнені обов’язкові поля; 4.2) система повідомляє користувача про необхідність заповнити обов’язкові поля.

Таблиця 1.13 – Варіант використання UC012

Назва	Видалення створеної вакансії.
Опис	Користувач з типом «Роботодавець» має можливість видалити свої вакансії.
Учасники	Користувач.
Передумови	Користувач авторизований в системі.
Постумови	Обрана вакансія видалена.
Основний сценарій	1) користувач обирає вакансію, яку хоче видалити; 2) користувач натискає кнопку “Видалити”; 3) система видаляє вакансію.
Розширення сценаріїв	

Таблиця 1.14 – Варіант використання UC013

Назва	Перегляд вакансій.
Опис	Користувач з типом «Роботодавець» має можливість переглядати свої вакансії.
Учасники	Користувач.
Передумови	Користувач авторизований в системі.
Постумови	Надано список створених користувачем вакансій.
Основний сценарій	1) користувач відкриває список створених вакансій; 2) система відображає список створених користувачем вакансій.
Розширення сценаріїв	

Таблиця 1.15 – Варіант використання UC014

Назва	Авторизація адміністратора.
Опис	Адміністратор має можливість авторизуватись в системі.
Учасники	Адміністратор.
Передумови	
Постумови	Адміністратор перенаправляється до особистого кабінету.
Основний сценарій	1) система відображає вікно авторизації, яке містить логін та пароль; 2) адміністратор заповнює поля вводу; 3) адміністратор натискає кнопку “Вхід”;

## Продовження таблиці 1.15

	4) система авторизує адміністратора;
Розширення сценаріїв	4.1) система виявляє, що Адміністратор ввів некоректні данні; 4.1.а) система демонструє повідомлення на екрані, яку саме помилку допустив користувач.

## Таблиця 1.16 – Варіант використання UC015

Назва	Відображення міст.
Опис	Адміністратор має можливість переглядати міста, які може вибирати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Надано список міст, які може вибирати користувач.
Основний сценарій	1) адміністратор переходить на сторінку «Міста»; 2) система відображає список міст, які може вибирати користувач.
Розширення сценаріїв	

## Таблиця 1.17 – Варіант використання UC016

Назва	Додавання міста.
Опис	Адміністратор має можливість додавати міста, які може вибирати користувач.

Продовження таблиці 1.17

Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Місто додано в таблицю міст.
Основний сценарій	1) адміністратор переходить на сторінку «Міста»; 2) адміністратор заповнює обов'язкові поля «Місто», «City»; 3) адміністратор натискає на кнопку «Зберегти»; 4) система додає місто в таблицю міст;
Розширення сценаріїв	4.1) система перевіряє чи заповнені обов'язкові поля; 4.2) система повідомляє користувача про необхідність заповнити обов'язкові поля.

Таблиця 1.18 – Варіант використання UC017

Назва	Видалення міста.
Опис	Адміністратор має можливість видалити місто з таблиці міст.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Обране місто видалене.
Основний сценарій	1) адміністратор обирає місто, яке хоче видалити; 2) адміністратор натискає кнопку «Видалити»; 3) система видаляє місто.

## Продовження таблиці 1.18

Розширення сценаріїв	
----------------------	--

## Таблиця 1.19 – Варіант використання UC018

Назва	Відображення курсів.
Опис	Адміністратор має можливість переглядати курси, які можуть відображатися користувачу.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Надано список курсів, які можуть відображатися користувачу, та для якої мови програмування.
Основний сценарій	1) адміністратор переходить на сторінку «Курси»; 2) система відображає список курсів, які можуть відображатися користувачу, та для якої мови програмування.
Розширення сценаріїв	

## Таблиця 1.20 – Варіант використання UC019

Назва	Додавання курсу.
Опис	Адміністратор має можливість додавати курси, які можуть відображатися користувачу.
Учасники	Адміністратор.

## Продовження таблиці 1.20

Передумови	Адміністратор авторизований в системі.
Постумови	Курс додано в таблицю курсів.
Основний сценарій	1) адміністратор переходить на сторінку «Курси»; 2) адміністратор заповнює обов'язкові поля «Назва», «Посилання», та обирає до яких мов програмування він підходить; 3) адміністратор натискає на кнопку «Зберегти»; 4) система додає курс в таблицю курсів;
Розширення сценаріїв	4.1) система перевіряє чи заповнені обов'язкові поля; 4.2) система повідомляє користувача про необхідність заповнити обов'язкові поля.

## Таблиця 1.21 – Варіант використання UC020

Назва	Видалення курсу.
Опис	Адміністратор має можливість видалити курс з таблиці курсів.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Обраний курс видалено.
Основний сценарій	1) адміністратор обирає курс, який хоче видалити; 2) адміністратор натискає кнопку «Видалити»; 3) система видаляє курс.



## Продовження таблиці 1.21

Розширення сценаріїв	
----------------------	--

Таблиця 1.22 – Варіант використання UC021

Назва	Відображення рівнів освіти.
Опис	Адміністратор має можливість переглядати рівні освіти, які може вибрати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Надано список рівнів освіти які може вибрати користувач.
Основний сценарій	1) адміністратор переходить на сторінку «Рівні освіти»; 2) система відображає список рівнів освіти, які може вибрати користувач.
Розширення сценаріїв	

Таблиця 1.23 – Варіант використання UC022

Назва	Додавання рівнів освіти.
Опис	Адміністратор має можливість додавати рівні освіти, які може вибрати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.

## Продовження таблиці 1.23

Постумови	Рівень освіти додано в таблицю рівнів освіти.
Основний сценарій	1) адміністратор переходить на сторінку «Рівні освіти»; 2) адміністратор заповнює обов'язкове поле «Рівень освіти»; 3) адміністратор натискає на кнопку «Зберегти»; 4) система додає рівень освіти в таблицю рівнів освіти;
Розширення сценаріїв	4.1) система перевіряє чи заповнене обов'язкове поле; 4.2) система повідомляє користувача про необхідність заповнити обов'язкове поле.

## Таблиця 1.24 – Варіант використання UC023

Назва	Видалення рівня освіти.
Опис	Адміністратор має можливість видалити рівень освіти з таблиці рівнів освіти.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Обраний рівень освіти видалено.
Основний сценарій	1) адміністратор обирає рівень освіти, який хоче видалити; 2) адміністратор натискає кнопку «Видалити»; 3) система видаляє рівень освіти.
Розширення сценаріїв	

Таблиця 1.25 – Варіант використання UC024

Назва	Відображення типів зайнятості.
Опис	Адміністратор має можливість переглядати типи зайнятості, які може вибрати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Надано список типів зайнятості, які може вибрати користувач.
Основний сценарій	1) адміністратор переходить на сторінку «Типи зайнятості»; 2) система відображає список типів зайнятості, які може вибрати користувач.
Розширення сценаріїв	

Таблиця 1.26 – Варіант використання UC025

Назва	Додавання типів зайнятості.
Опис	Адміністратор має можливість додавати типи зайнятості, які може вибрати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Тип зайнятості додано в таблицю рівнів освіти.

## Продовження таблиці 1.26

Основний сценарій	1) адміністратор переходить на сторінку «Типи зайнятості»; 2) адміністратор заповнює обов'язкове поле «Тип зайнятості»; 3) адміністратор натискає на кнопку «Зберегти»; 4) система додає тип зайнятості в таблицю типів зайнятості;
Розширення сценаріїв	4.1) система перевіряє чи заповнене обов'язкове поле. 4.2) система повідомляє користувача про необхідність заповнити обов'язкове поле.

Таблиця 1.27 – Варіант використання UC026

Назва	Видалення типів зайнятості.
Опис	Адміністратор має можливість видалити тип зайнятості з таблиці типів зайнятості.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Обраний тип зайнятості видалено.
Основний сценарій	1) адміністратор обирає тип зайнятості, який хоче видалити; 2) адміністратор натискає кнопку «Видалити»; 3) система видаляє тип зайнятості.
Розширення сценаріїв	

Таблиця 1.28 – Варіант використання UC027

Назва	Відображення рівнів навичок.
Опис	Адміністратор має можливість переглядати рівні навичок, які може вибирати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Надано список рівнів навичок, які може вибирати користувач.
Основний сценарій	1) адміністратор переходить на сторінку «Рівень навичок»; 2) система відображає список рівнів навичок, які може вибирати користувач.
Розширення сценаріїв	

Таблиця 1.29 – Варіант використання UC028

Назва	Додавання рівнів навичок.
Опис	Адміністратор має можливість додавати рівні навичок, які може вибирати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Рівень навичок додано в таблицю рівнів освіти.

## Продовження таблиці 1.29

Основний сценарій	1) адміністратор переходить на сторінку «Рівень навичок»; 2) адміністратор заповнює обов'язкове поле «Рівень навичка»; 3) адміністратор натискає на кнопку «Зберегти»; 4) система додає рівень навичка в таблицю рівні зайнятості;
Розширення сценаріїв	4.1) система перевіряє чи заповнене обов'язкове поле; 4.2) система повідомляє користувача про необхідність заповнити обов'язкове поле.

Таблиця 1.30 – Варіант використання UC029

Назва	Видалення рівня навичок.
Опис	Адміністратор має можливість видалити рівень навичок з таблиці типів зайнятості.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Обраний рівень навичка видалено.
Основний сценарій	1) адміністратор обирає рівень навичка, який хоче видалити; 2) адміністратор натискає кнопку «Видалити»; 3) система видаляє рівень навичка.
Розширення сценаріїв	

Таблиця 1.31 – Варіант використання UC030

Назва	Відображення рубрик.
Опис	Адміністратор має можливість переглядати рубрики, які може вибрати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Надано список рубрик, які може вибрати користувач.
Основний сценарій	1) адміністратор переходить на сторінку «Рубрики»; 2) система відображає список рубрик, які може вибрати користувач.
Розширення сценаріїв	

Таблиця 1.32 – Варіант використання UC031

Назва	Додавання рубрик.
Опис	Адміністратор має можливість додавати рубрики, які може вибрати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Рубрика додана в таблицю рубрик.

## Продовження таблиці 1.32

Основний сценарій	1) адміністратор переходить на сторінку «Рубрики».; 2) адміністратор заповнює обов'язкове поле «Рубрика»; 3) адміністратор натискає на кнопку «Зберегти»; 4) система додає рубрику в таблицю рубрики;
Розширення сценаріїв	4.1) система перевіряє чи заповнене обов'язкове поле; 4.2) система повідомляє користувача про необхідність заповнити обов'язкове поле.

Таблиця 1.33 – Варіант використання UC032

Назва	Видалення рубрики.
Опис	Адміністратор має можливість видалити рубрику з таблиці рубрики.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Обрана рубрика видалена.
Основний сценарій	1) адміністратор обирає рубрику , яку хоче видалити; 2) адміністратор натискає кнопку «Видалити»; 3) система видаляє рубрику.
Розширення сценаріїв	



Таблиця 1.34 – Варіант використання UC033

Назва	Відображення сфер діяльності.
Опис	Адміністратор має можливість переглядати сфери діяльності, які може вибирати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Надано список сфер діяльності, які може вибирати користувач.
Основний сценарій	1) адміністратор переходить на сторінку «Сфери діяльності»; 2) система відображає список сфер діяльності, які може вибирати користувач.
Розширення сценаріїв	

Таблиця 1.35 – Варіант використання UC034

Назва	Додавання сфер діяльності.
Опис	Адміністратор має можливість додавати сфери діяльності, які може вибирати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Сфера діяльності додана в таблицю сфери діяльності.

## Продовження таблиці 1.35

Основний сценарій	1) адміністратор переходить на сторінку «Сфери діяльності»; 2) адміністратор заповнює обов'язкове поле «Сфера діяльності»; 3) адміністратор натискає на кнопку «Зберегти»; 4) система додає сферу діяльності в таблицю сфер діяльності;
Розширення сценаріїв	4.1) система перевіряє чи заповнене обов'язкове поле; 4.2) система повідомляє користувача про необхідність заповнити обов'язкове поле.

Таблиця 1.36 – Варіант використання UC035

Назва	Видалення сфери діяльності.
Опис	Адміністратор має можливість видалити сферу діяльності з таблиці сфери діяльності.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Обрана сфера діяльності видалена.
Основний сценарій	1) адміністратор обирає сферу діяльності, яку хоче видалити; 2) адміністратор натискає кнопку «Видалити»; 3) система видаляє сферу діяльності.
Розширення сценаріїв	

Таблиця 1.37 – Варіант використання UC036

Назва	Відображення навичок.
Опис	Адміністратор має можливість переглядати навички, які може вибрати користувач, та рубрики яким вони належать.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Надано навички, які може вибрати користувач.
Основний сценарій	1) адміністратор переходить на сторінку «Навички»; 2) система відображає список навичок, які може вибрати користувач.
Розширення сценаріїв	

Таблиця 1.38 – Варіант використання UC037

Назва	Додавання навички.
Опис	Адміністратор має можливість додавати навички, які може вибрати користувач, та вибрати рубрику якій воно відповідає.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Навичка додана в таблицю навичок.

## Продовження таблиці 1.38

Основний сценарій	1) адміністратор переходить на сторінку «Навички»; 2) адміністратор заповнює обов'язкове поле «Назва»; 3) адміністратор вибирає до якої рубрики відноситься навичка; 4) адміністратор натискає на кнопку “Зберегти”; 5) система додає навичку в таблицю навичок;
Розширення сценаріїв	5.1) система перевіряє чи заповнене обов'язкове поле; 5.2) система повідомляє користувача про необхідність заповнити обов'язкове поле.

Таблиця 1.39 – Варіант використання UC038

Назва	Видалення навички.
Опис	Адміністратор має можливість видалити навичку з таблиці навички.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Обрана навичка видалена.
Основний сценарій	1) адміністратор обирає навичку, яку хоче видалити; 2) адміністратор натискає кнопку “Видалити”; 3) система видаляє навичку.
Розширення сценаріїв	

Таблиця 1.40 – Варіант використання UC039

Назва	Відображення університетів.
Опис	Адміністратор має можливість переглядати університети, які може вибрати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Надано університети, які може вибрати користувач.
Основний сценарій	1) адміністратор переходить на сторінку «Університети»; 2) система відображає список університетів, які може вибрати користувач.
Розширення сценаріїв	

Таблиця 1.41 – Варіант використання UC040

Назва	Додавання університетів.
Опис	Адміністратор має можливість додавати університети, які може вибрати користувач.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Університет додано в таблицю університети.

## Продовження таблиці 1.41

Основний сценарій	1) адміністратор переходить на сторінку «Університети»; 2) адміністратор заповнює обов'язкове поле «Університет»; 3) адміністратор натискає на кнопку «Зберегти»; 4) система додає університет в таблицю університети;
Розширення сценаріїв	4.1) система перевіряє чи заповнене обов'язкове поле; 4.2) система повідомляє користувача про необхідність заповнити обов'язкове поле.

Таблиця 1.42 – Варіант використання UC041

Назва	Видалення університету.
Опис	Адміністратор має можливість видалити університет з таблиці університети.
Учасники	Адміністратор.
Передумови	Адміністратор авторизований в системі.
Постумови	Обраний університет видалений.
Основний сценарій	1) адміністратор обирає університет, який хоче видалити; 2) адміністратор натискає кнопку «Видалити»; 3) система видаляє університет.
Розширення сценаріїв	

Таблиця 1.43 – Варіант використання UC042

Назва	Вихід користувача з особистого кабінету.
Опис	Користувач може вийти з особистого кабінету.
Учасники	Користувач.
Передумови	Користувач авторизувався в системі.
Постумови	Користувач вийшов з системи.
Основний сценарій	Користувач натискає кнопку “Вихід”, яка знаходиться у верхній частині екрану.
Розширення сценаріїв	

Таблиця 1.44 – Варіант використання UC043

Назва	Вихід адміністратора з особистого кабінету
Опис	Адміністратор може вийти з особистого кабінету.
Учасники	Адміністратор.
Передумови	Адміністратор авторизувався в системі.
Постумови	Адміністратор вийшов з системи.
Основний сценарій	Адміністратор натискає кнопку “Вихід”, яка знаходиться у верхній частині екрану.
Розширення сценаріїв	

Дипломний проект має такі функціональні вимоги:

Таблиця 1.45 – Опис функціональної вимоги REQ001

Номер	REQ001
Назва	Реєстрація користувачів в системі.
Опис	Користувач має можливість зареєструватися в системі.

Таблиця 1.46 – Опис функціональної вимоги REQ002

Номер	REQ002
Назва	Авторизація користувачів в системі.
Опис	Користувач має можливість авторизуватися в системі.

Таблиця 1.47 – Опис функціональної вимоги REQ003

Номер	REQ003
Назва	Створення резюме.
Опис	Користувач має можливість створювати резюме.

Таблиця 1.48 – Опис функціональної вимоги REQ004

Номер	REQ004
Назва	Перегляд списку резюме.
Опис	Користувач має можливість переглядати список своїх резюме.



Таблиця 1.49 – Опис функціональної вимоги REQ005

Номер	REQ005
Назва	Редагування резюме.
Опис	Користувач має можливість редагувати свої резюме.

Таблиця 1.50 – Опис функціональної вимоги REQ006

Номер	REQ006
Назва	Видалення резюме.
Опис	Користувач має можливість видаляти свої резюме.

Таблиця 1.51 – Опис функціональної вимоги REQ007

Номер	REQ007
Назва	Створення вакансій.
Опис	Користувач має можливість створювати вакансії.

Таблиця 1.52 – Опис функціональної вимоги REQ008

Номер	REQ008
Назва	Перегляд списку своїх вакансій.
Опис	Користувач має можливість переглядати список своїх вакансій.

Таблиця 1.53 – Опис функціональної вимоги REQ009

Номер	REQ009
Назва	Редагування вакансій.
Опис	Користувач має можливість редагувати свої вакансії.

Таблиця 1.54 – Опис функціональної вимоги REQ010

Номер	REQ010
Назва	Видалення вакансій.
Опис	Користувач має можливість видаляти свої вакансії.

Таблиця 1.55 – Опис функціональної вимоги REQ011

Номер	REQ011
Назва	Пошук курсів.
Опис	Користувач має можливість шукати курси по різних мовам програмування.

Таблиця 1.56 – Опис функціональної вимоги REQ012

Номер	REQ012
Назва	Перегляд статистики.
Опис	Користувач має можливість переглядати статистику, щодо кількості вакансій на сайтах пошуку роботи по різних мовам програмування.

Таблиця 1.57 – Опис функціональної вимоги REQ013

Номер	REQ013
Назва	Оновлення статистики.
Опис	Користувач має можливість оновити статистику, щодо кількості вакансій на сайтах пошуку роботи по різним мовам програмування.

Таблиця 1.58 – Опис функціональної вимоги REQ014

Номер	REQ014
Назва	Авторизація адміністратора в системі.
Опис	Адміністратор має можливість авторизуватися в системі.

Таблиця 1.59 – Опис функціональної вимоги REQ015

Номер	REQ015
Назва	Додавання міст.
Опис	Адміністратор має можливість додавати міста в таблицю міст.

Таблиця 1.60 – Опис функціональної вимоги REQ016

Номер	REQ016
Назва	Перегляд списку міст.
Опис	Адміністратор має можливість переглядати список міст з таблиці міст.

Таблиця 1.61 – Опис функціональної вимоги REQ017

Номер	REQ017
Назва	Видалення міста.
Опис	Адміністратор має можливість видаляти міста з таблиці міст.

Таблиця 1.62 – Опис функціональної вимоги REQ018

Номер	REQ018
Назва	Додавання курсу.
Опис	Адміністратор має можливість додавати курси в таблицю курсів.

Таблиця 1.63 – Опис функціональної вимоги REQ019

Номер	REQ019
Назва	Перегляд списку курсів.
Опис	Адміністратор має можливість переглядати список курсів з таблиці курсів.

Таблиця 1.64 – Опис функціональної вимоги REQ020

Номер	REQ020
Назва	Видалення курсу.
Опис	Адміністратор має можливість видаляти курс з таблиці курсів.

Таблиця 1.65 – Опис функціональної вимоги REQ021

Номер	REQ021
Назва	Додавання рівня освіти.
Опис	Адміністратор має можливість додавати рівень освіти в таблицю рівнів освіти.

Таблиця 1.66 – Опис функціональної вимоги REQ022

Номер	REQ022
Назва	Перегляд списку рівнів освіти.
Опис	Адміністратор має можливість переглядати список рівнів освіти з таблиці рівнів освіти.

Таблиця 1.67 – Опис функціональної вимоги REQ023

Номер	REQ023
Назва	Видалення рівнів освіти.
Опис	Адміністратор має можливість видаляти рівень освіти з таблиці рівнів освіти.

Таблиця 1.68 – Опис функціональної вимоги REQ024

Номер	REQ024
Назва	Додавання типів зайнятості.
Опис	Адміністратор має можливість додавати тип зайнятості в таблицю типів зайнятості.

Таблиця 1.69 – Опис функціональної вимоги REQ025

Номер	REQ025
Назва	Перегляд списку типів зайнятості.
Опис	Адміністратор має можливість переглядати список типів зайнятості з таблиці типів зайнятості.

Таблиця 1.70 – Опис функціональної вимоги REQ026

Номер	REQ026
Назва	Видалення типів зайнятості.
Опис	Адміністратор має можливість видаляти тип зайнятості з таблиці типів зайнятості.

Таблиця 1.71 – Опис функціональної вимоги REQ027

Номер	REQ027
Назва	Додавання рівнів навичок.
Опис	Адміністратор має можливість додавати рівень навичок в таблицю рівнів навичок.

Таблиця 1.72 – Опис функціональної вимоги REQ028

Номер	REQ028
Назва	Перегляд списку рівнів навичок.
Опис	Адміністратор має можливість переглядати список рівнів навичок з таблиці рівнів навичок.

Таблиця 1.73 – Опис функціональної вимоги REQ029

Номер	REQ029
Назва	Видалення рівнів навичок.
Опис	Адміністратор має можливість видаляти рівень навичок з таблиці рівнів навичок.

Таблиця 1.74 – Опис функціональної вимоги REQ030

Номер	REQ030
Назва	Додавання рубрик.
Опис	Адміністратор має можливість додавати рубрику в таблицю рубрик.

Таблиця 1.75 – Опис функціональної вимоги REQ031

Номер	REQ031
Назва	Перегляд списку рубрик.
Опис	Адміністратор має можливість переглядати список рубрик з таблиці рубрик.

Таблиця 1.76 – Опис функціональної вимоги REQ032

Номер	REQ032
Назва	Видалення рубрик.
Опис	Адміністратор має можливість видаляти рубрику з таблиці рубрик.

Таблиця 1.77 – Опис функціональної вимоги REQ033

Номер	REQ033
Назва	Додавання сфер діяльності.
Опис	Адміністратор має можливість додавати сферу діяльності в таблицю сфер діяльності.

Таблиця 1.78 – Опис функціональної вимоги REQ034

Номер	REQ034
Назва	Перегляд списку сфер діяльності.
Опис	Адміністратор має можливість переглядати список сфер діяльності з таблиці сфер діяльності.

Таблиця 1.79 – Опис функціональної вимоги REQ035

Номер	REQ035
Назва	Видалення сфер діяльності.
Опис	Адміністратор має можливість видаляти сферу діяльності з таблиці сфер діяльності.

Таблиця 1.80 – Опис функціональної вимоги REQ036

Номер	REQ036
Назва	Додавання навичок.
Опис	Адміністратор має можливість додавати навички в таблицю навичок.



Таблиця 1.81 – Опис функціональної вимоги REQ037

Номер	REQ037
Назва	Перегляд списку навичок.
Опис	Адміністратор має можливість переглядати список навичок з таблиці навичок.

Таблиця 1.82 – Опис функціональної вимоги REQ038

Номер	REQ038
Назва	Видалення навичок.
Опис	Адміністратор має можливість видаляти навички з таблиці навичок.

Таблиця 1.83 – Опис функціональної вимоги REQ039

Номер	REQ039
Назва	Додавання університетів.
Опис	Адміністратор має можливість додавати університет в таблицю університетів.

Таблиця 1.84 – Опис функціональної вимоги REQ040

Номер	REQ040
Назва	Перегляд списку університетів.
Опис	Адміністратор має можливість переглядати список університетів з таблиці університетів.

Таблиця 1.85 – Опис функціональної вимоги REQ041

Номер	REQ041
Назва	Видалення університетів.
Опис	Адміністратор має можливість видаляти університет з таблиці університетів.

Таблиця 1.86 – Опис функціональної вимоги REQ042

Номер	REQ042
Назва	Вихід користувача з особистого кабінету.
Опис	Система дозволяє користувачу вийти з особистого кабінету.

Таблиця 1.87 – Опис функціональної вимоги REQ043

Номер	REQ043
Назва	Вихід адміністратора з особистого кабінету.
Опис	Система дозволяє адміністратору вийти з особистого кабінету.

REQ001	Авторизація користувача	UC001	Авторизація користувача
REQ002	Рестрація користувача	UC002	Рестрація користувача
REQ003	Створення резюме	UC003	Створення резюме
REQ004	Перегляд списку резюме	UC004	Перегляд списку резюме
REQ005	Редагування резюме	UC005	Редагування резюме
REQ006	Видалення резюме	UC006	Видалення резюме
REQ007	Створення вакансій	UC007	Створення вакансій
REQ008	Перегляд списку своїх вакансій	UC008	Перегляд списку своїх вакансій
REQ009	Редагування вакансій	UC009	Редагування вакансій
REQ010	Видалення вакансій	UC010	Видалення вакансій
REQ011	Пошук курсів	UC011	Пошук курсів
REQ012	Перегляд статистики	UC012	Перегляд статистики
REQ013	Оновлення статистики	UC013	Оновлення статистики
REQ014	Авторизація адміністратора в системі	UC014	Авторизація адміністратора
REQ015	Додавання міст	UC015	Додавання міст
REQ016	Перегляд списку міст	UC016	Перегляд списку міст
REQ017	Видалення міста	UC017	Видалення міста
REQ018	Додавання курсу	UC018	Додавання курсу
REQ019	Перегляд списку курсів	UC019	Перегляд списку курсів
REQ020	Видалення курсу	UC020	Видалення курсу
REQ021	Додавання рівня освіти	UC021	Додавання рівня освіти
REQ022	Перегляд списку рівнів освіти	UC022	Перегляд списку рівнів освіти
REQ023	Видалення рівня освіти	UC023	Видалення рівня освіти
REQ024	Додавання типів зайнятості	UC024	Додавання типів зайнятості
REQ025	Перегляд списку типів зайнятості	UC025	Перегляд списку типів зайнятості
REQ026	Видалення типів зайнятості	UC026	Видалення типів зайнятості
REQ027	Додавання рівнів навичок	UC027	Додавання рівнів навичок
REQ028	Перегляд списку рівнів навичок	UC028	Перегляд списку рівнів навичок
REQ029	Видалення рівнів навичок	UC029	Видалення рівнів навичок
REQ030	Додавання рубрик	UC030	Додавання рубрик
REQ031	Перегляд списку рубрик	UC031	Перегляд списку рубрик
REQ032	Видалення рубрик	UC032	Видалення рубрик
REQ033	Додавання сфер діяльності	UC033	Додавання сфер діяльності
REQ034	Перегляд списку сфер діяльності	UC034	Перегляд списку сфер діяльності
REQ035	Видалення сфер діяльності	UC035	Видалення сфер діяльності
REQ036	Додавання навичок	UC036	Додавання навичок
REQ037	Перегляд списку навичок	UC037	Перегляд списку навичок
REQ038	Видалення навичок	UC038	Видалення навичок
REQ039	Додавання університетів	UC039	Додавання університетів
REQ040	Перегляд списку університетів	UC040	Перегляд списку університетів
REQ041	Видалення університетів	UC041	Видалення університетів
REQ042	Вихід користувача з особистого кабінету	UC042	Вихід користувача з особистого кабінету
REQ043	Вихід адміністратора з особистого кабінету	UC043	Вихід адміністратора з особистого кабінету

Рисунок 1.1 – Матриця трасування

#### 1.4.2 Розроблення нефункціональних вимог

Дипломний проект повинен відповідати наступним нефункціональним вимогам:

- інтерфейс повинен мати українську локалізацію;
- підтримка на всіх сучасних браузерях;
- інтерфейс зрозумілий для користувача;
- навантаження на систему залежить від ресурсів сервера;
- забезпечення надійності системи реалізується регулярним резервним копіюванням бази даних;
- дипломний проект легко перенести на інший сервер.

#### 1.4.3 Постановка комплексу завдань модулю

Цілями розробки є:

- полегшення створення резюме;
- спрощення пошуку роботи;
- відображення статистики популярності мов програмування та позицій у вакансіях.

Для досягнення поставлених цілей необхідно вирішити наступні завдання:

- реєстрація користувача (роботодавця чи пошукача);
- авторизація користувача;
- створення резюме;
- перегляд резюме;
- редагування резюме;
- формування резюме в PDF форматі;
- видалення резюме;
- створення вакансії;
- перегляд вакансії;

- редагування вакансії;
- видалення вакансії;
- пошук вакансій для резюме;
- пошук статей по темам вказаних навичок в резюме;
- перегляд доступних курсів по заданим навичкам;
- формування статистичних діаграм за позиціями та мовами програмування.

### 1.5 Висновки по розділу

У цьому розділі був здійснений опис та аналіз предметної області розробки. Описано та проаналізовано декілька програмних продуктів подібних до дипломного проекту. Визначено призначення розробки даного застосування. Наведені цілі та задачі розробки.

## 2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Моделювання та аналіз програмного забезпечення

Для моделювання та аналізу програмного забезпечення, розглянемо основні процеси використання проекту користувачем за допомогою BPMN діаграм.

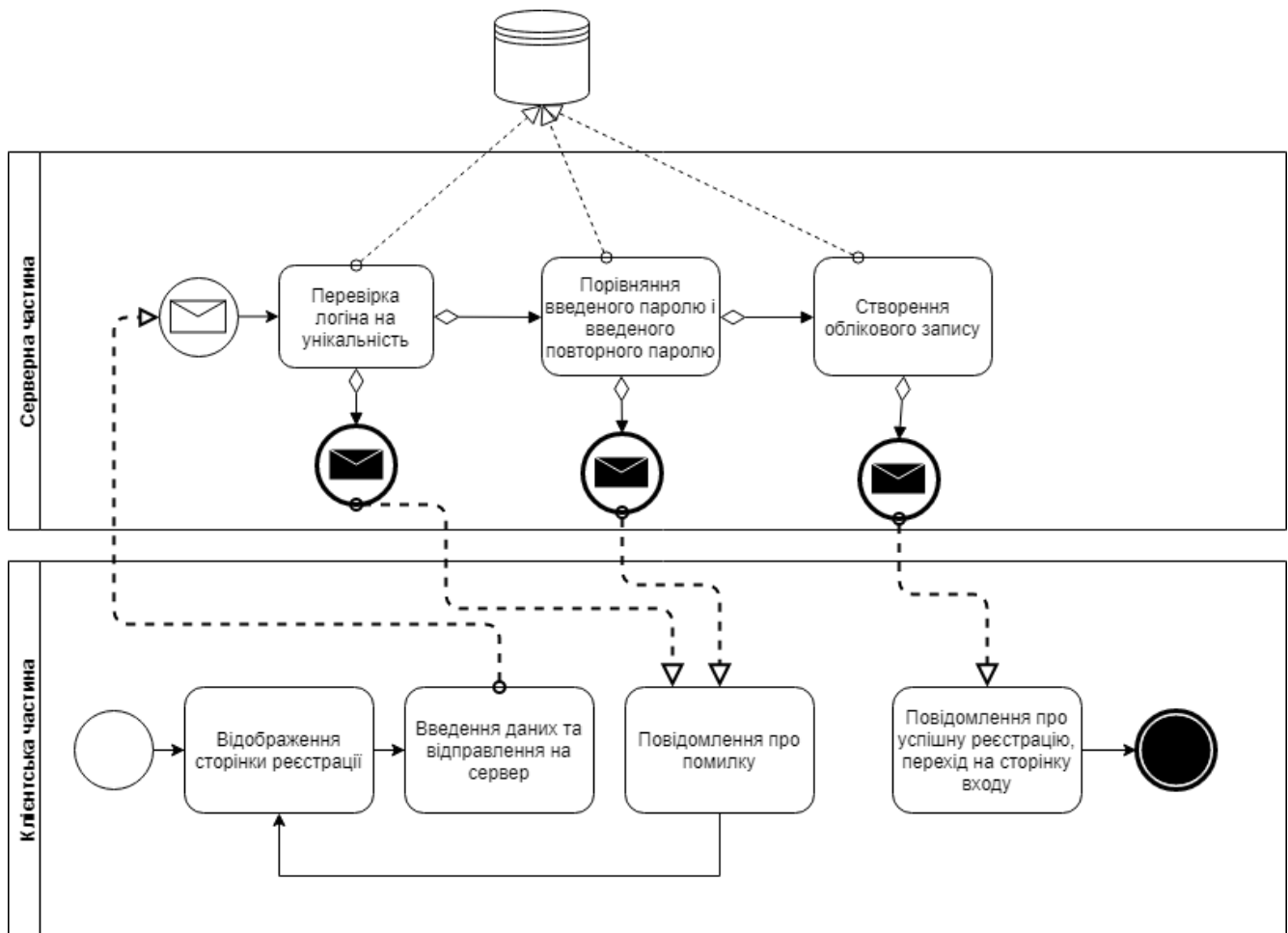


Рисунок 2.1 – Реєстрація користувача

Послідовний опис реєстрації користувача:

- користувачу відображується сторінка реєстрації;
- користувач заповнює поля для реєстрації;
- відправляється запит на реєстрацію на сервер;
- якщо логін уже існує, то користувач повідомляється про це;

- якщо пароль і повторно введений пароль не співпадають, то користувач повідомляється про різні введені паролі;
- якщо помилок не виявлено, створюється обліковий запис, та відображається повідомлення про це.

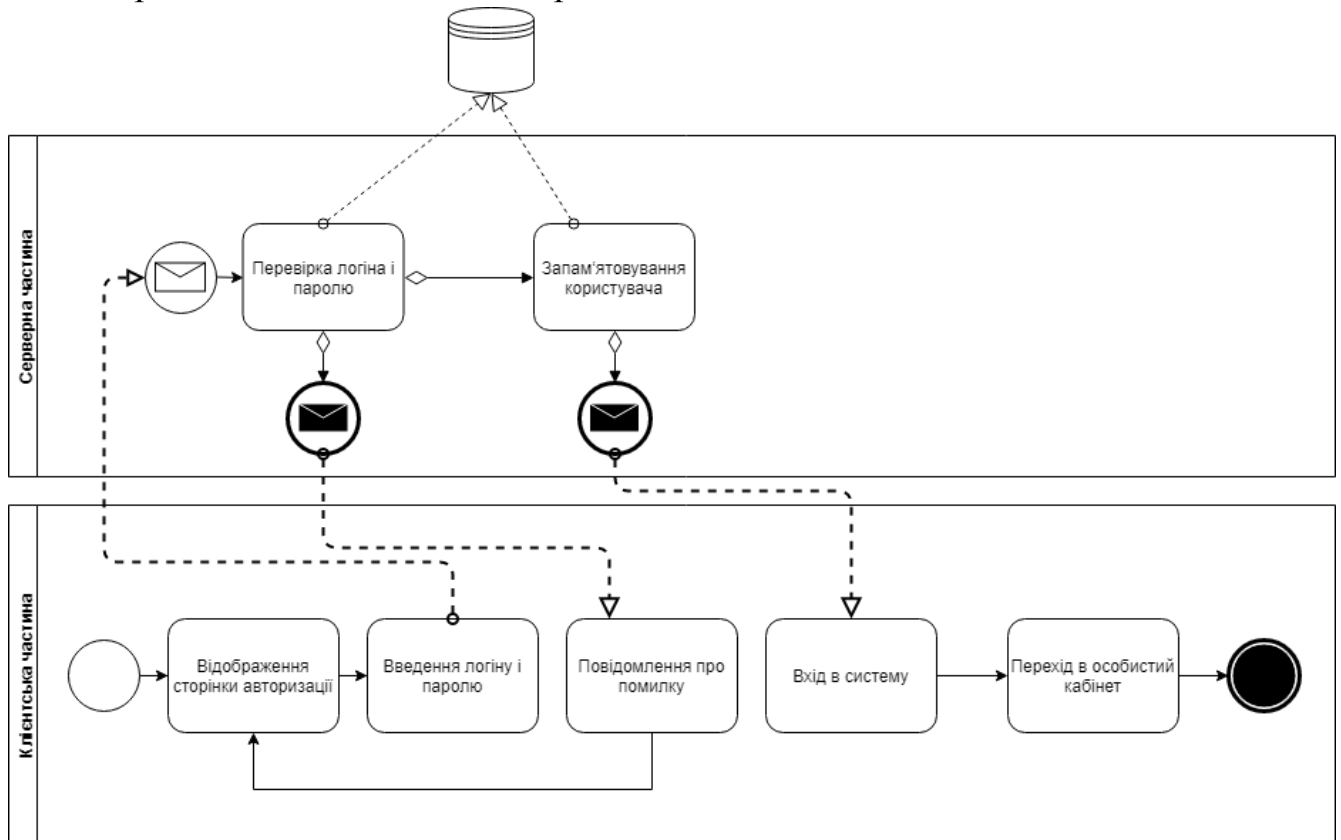


Рисунок 2.2 – Вхід користувача на сайт

Послідовний опис входу користувача на сайт:

- користувач заходить на сторінку авторизації;
- вводить логін і пароль;
- відбувається відправлення даних на сервер;
- якщо вони коректні – то сесія користувача запам'ятовується і він перенаправляється до особистого кабінету;
- якщо дані некоректні, то користувач повідомляється про помилку.

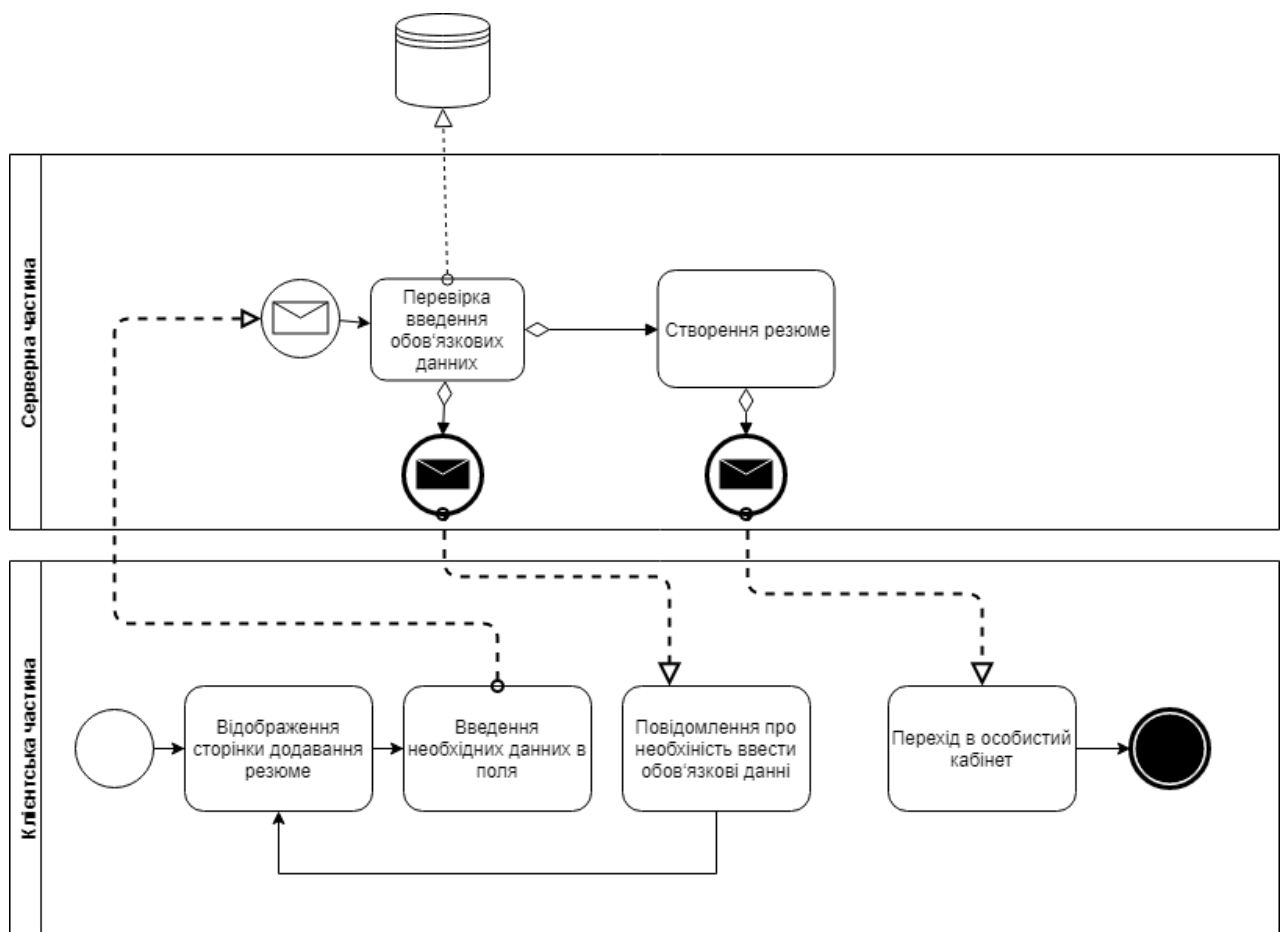


Рисунок 2.3 – Створення резюме

Послідовний опис створення резюме:

- користувач входить на сторінку додавання резюме;
- заповнює необхідні поля;
- відправляє на сервер запит на створення резюме;
- якщо всі необхідні поля заповнені, то сервер перенаправляє користувача в особистий кабінет;
- інакше повідомляє користувача про необхідність заповнити обов'язкові поля.



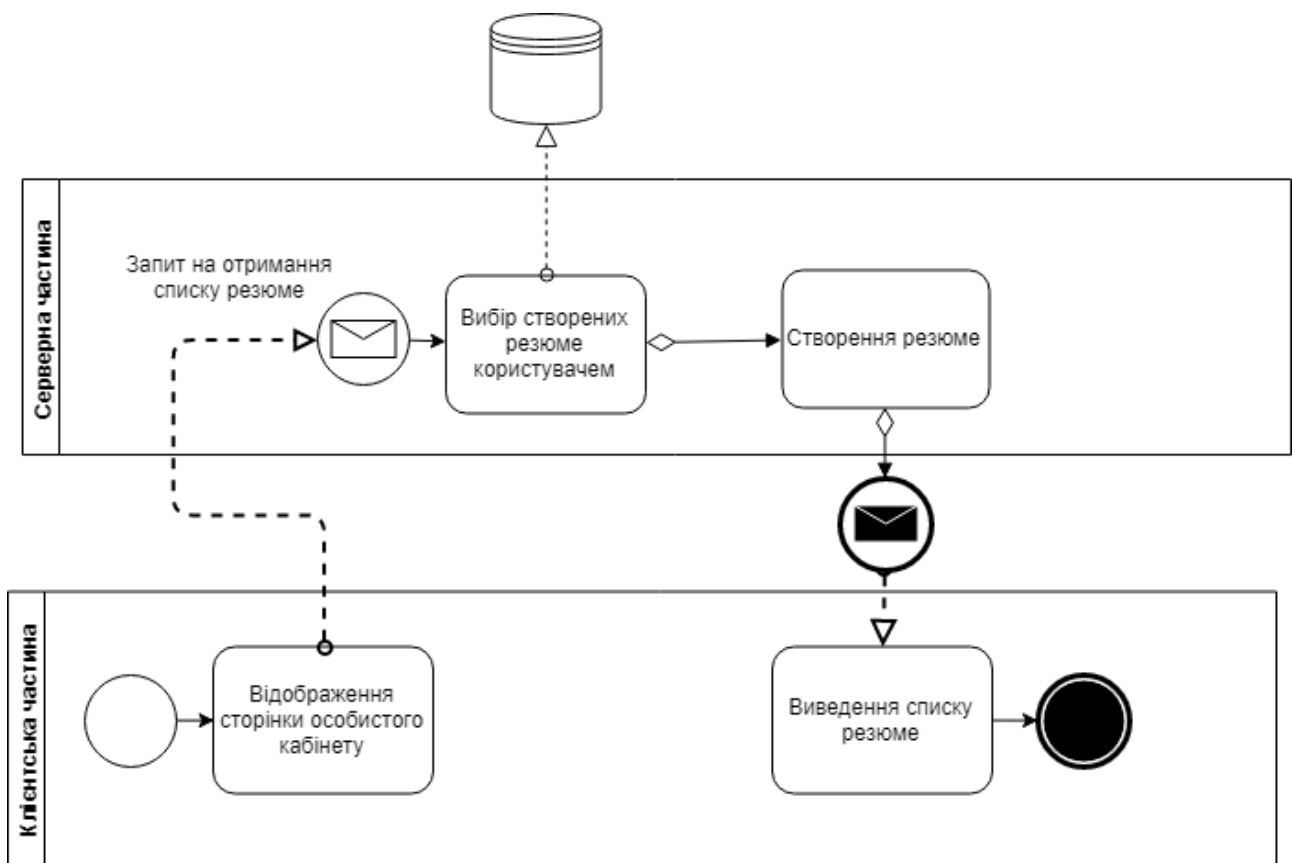


Рисунок 2.4 – Отримання списку резюме

Послідовний опис отримання списку резюме:

- користувач переходить до особистого кабінету, і автоматично на сервер іде запит на отримання списку резюме;
- сервер вибирає створені користувачем резюме;
- сервер надсилає список резюме.

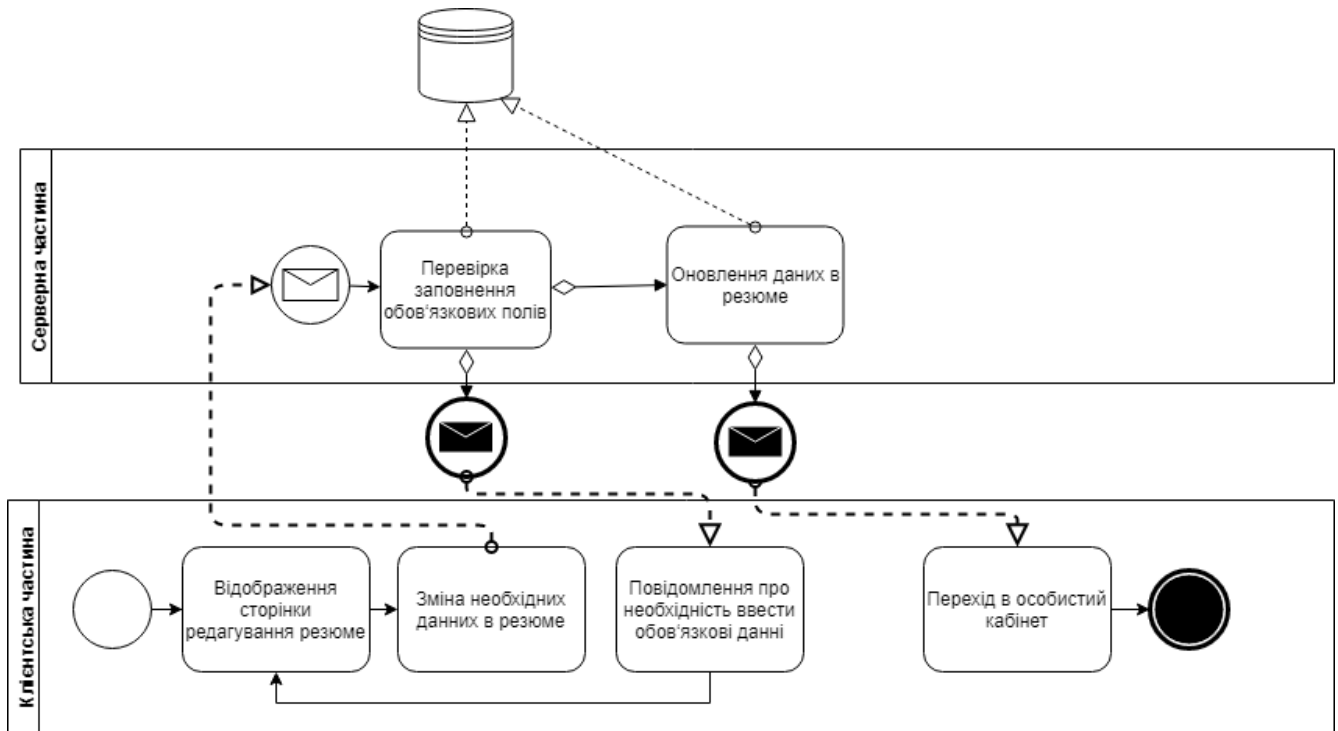


Рисунок 2.5 – Редагування резюме

Послідовний опис редагування резюме:

- користувач входить на сторінку редагування резюме;
- змінює необхідні дані в резюме;
- якщо всі необхідні поля заповнені, то сервер оновлює питання, і перенаправляє до особистого кабінету;
- інакше повідомляє користувача про необхідність заповнити обов'язкові поля.

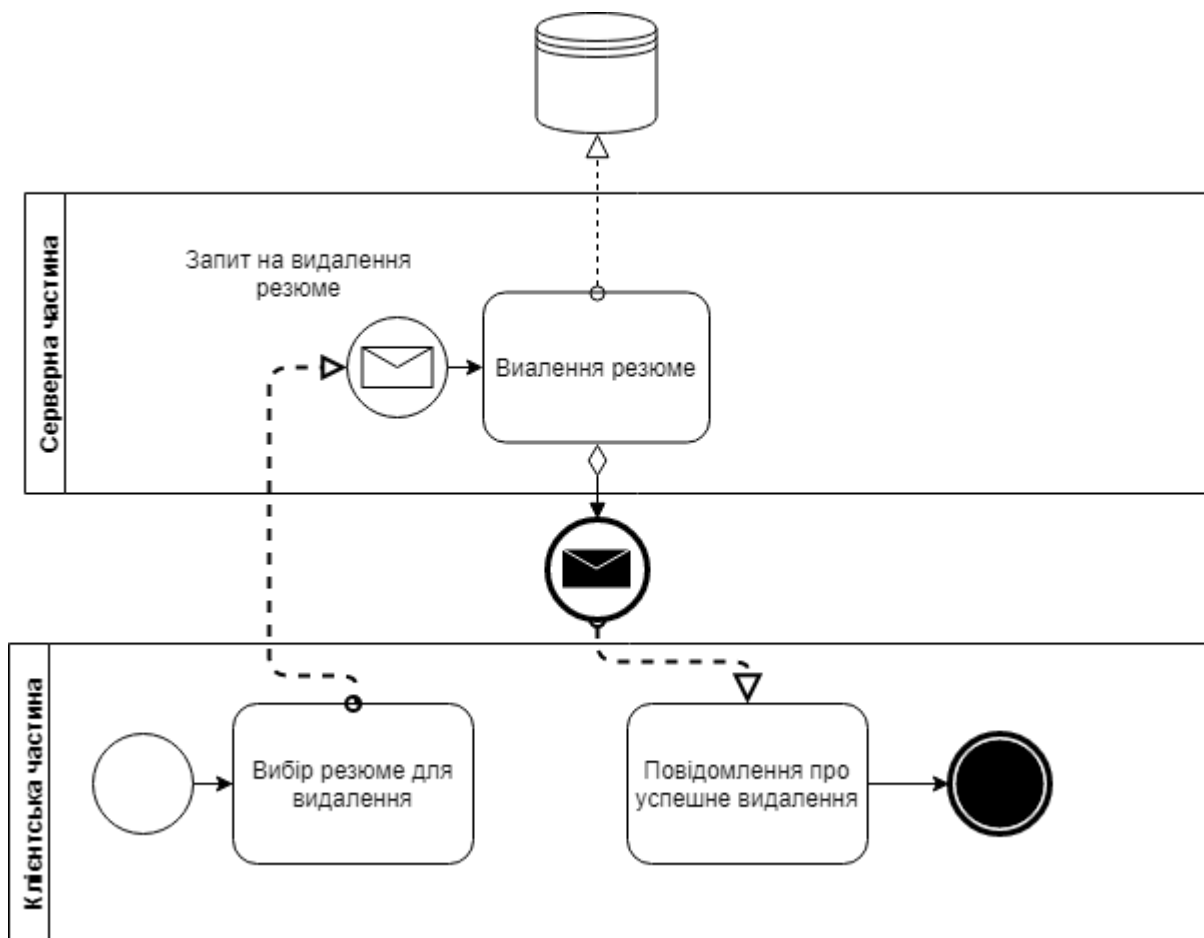


Рисунок 2.6 – Видалення резюме

Послідовний опис видалення резюме:

- користувач обирає резюме для видалення;
- сервер видаляє резюме з бази даних;
- сервер відправляє повідомлення про видалення резюме.

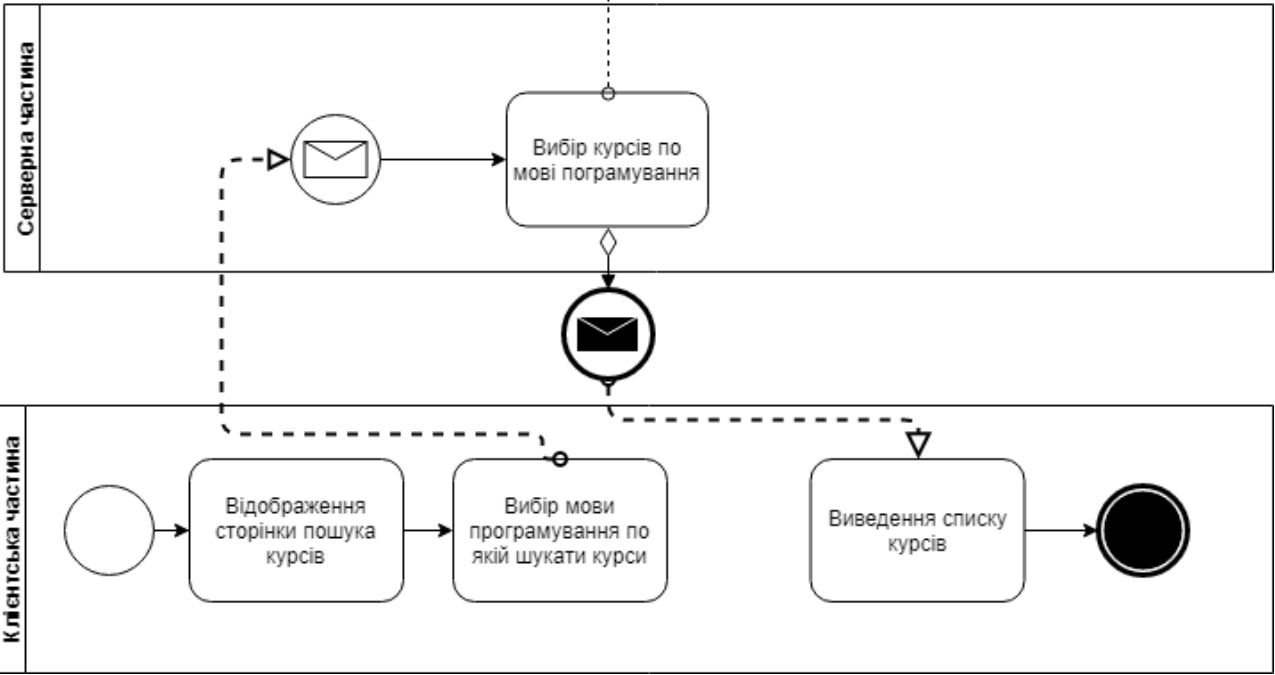


Рисунок 2.7 – Пошук курсів

## Послідовний опис пошуку курсів:

- користувач переходить на сторінку пошуку курсів;
- вибирає мову програмування по якій шукати курси;
- сервер вибирає курси по обраній мові програмування;
- сервер виводить список курсів.

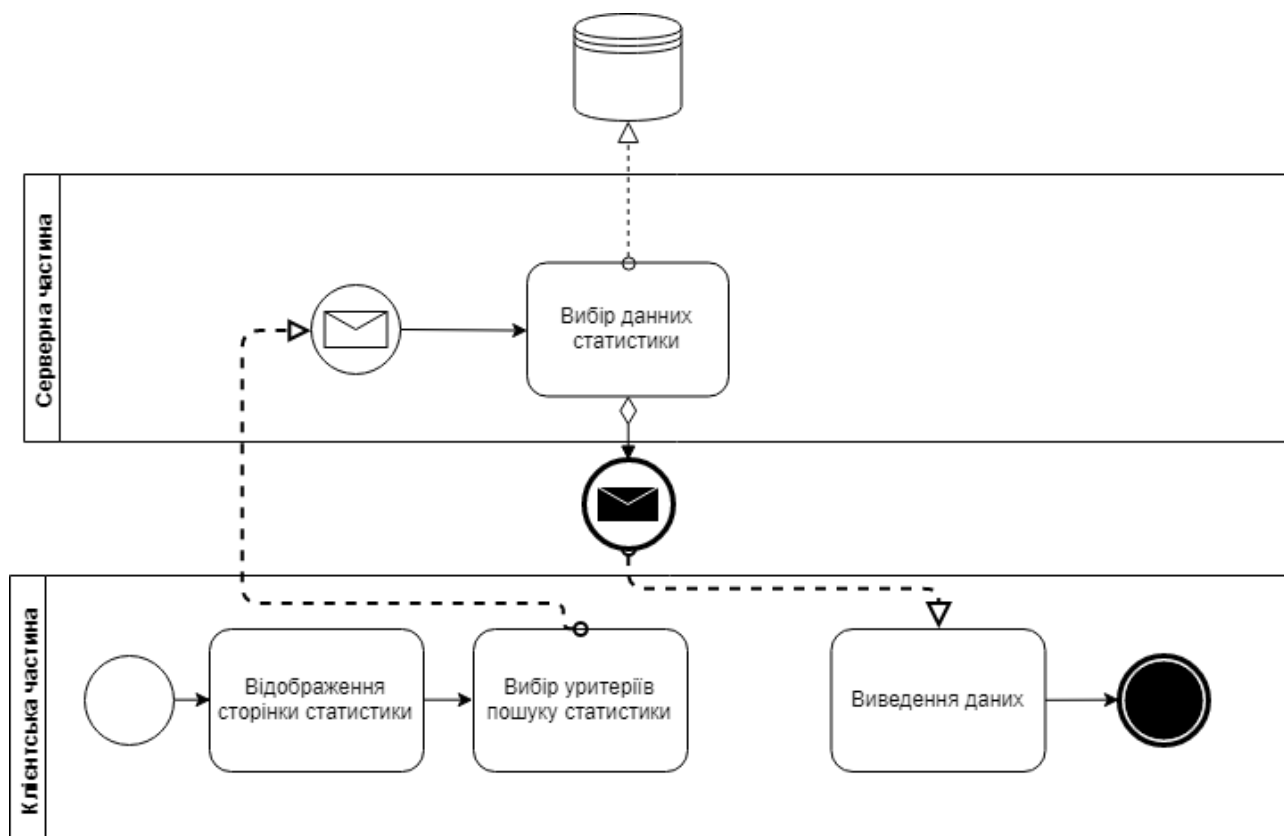


Рисунок 2.8 – Показ статистики

Послідовний опис показу статистики:

- користувач переходить на сторінку статистики;
- вибирає по якому критерію показувати статистику;
- сервер вибирає статистику по заданому критерію;
- сервер виводить статистику.

## 2.2 Архітектура програмного забезпечення

Загальна архітектура клієнт-серверного застосування показана на рисунку:

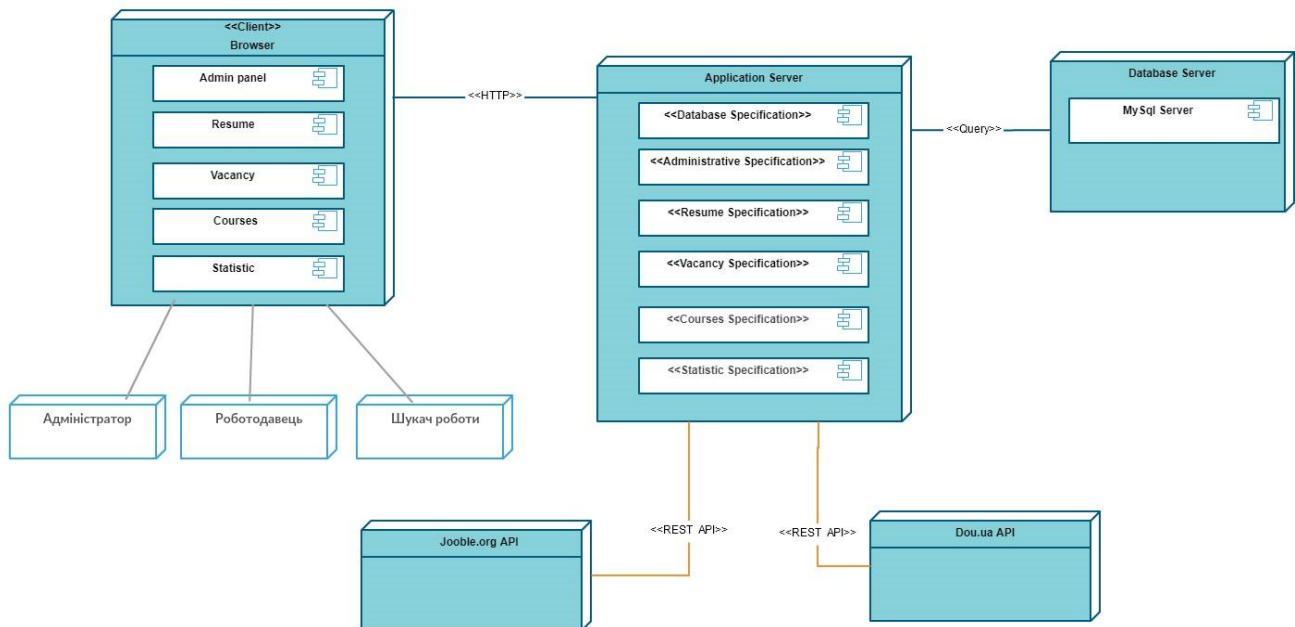


Рисунок 2.9 – Архітектура дипломного проекту

Для створення клієнтської частини була використана мова розмітки веб-сторінок HTML та використана бібліотека JQuery для надання динамічності сторінкам. Також для динамічної загрузки даних з сервера була використана технологія AJAX. Для розробки серверної частини веб-додатку була використана мова програмування PHP, оскільки вона добре поєднується з HTML тегами, і реляційна база даних MySQL була обрана з тієї ж причини.

При переході на сторінку спочатку PHP відправляє запит на отримання даних з бази даних, в базі даних береться потрібна інформація, формується розмітка з даними взятими з бази даних та відображається клієнту.

При роботі з API на API посилається запит в форматі JSON, з якого повертається відповідь також у форматі JSON, яка обробляється на сервері і відображається клієнту в зручному вигляді.

Опис структури бази даних наведено в додатку, нижче описані основні таблиці бази даних:

Таблиця 2.1 – *Jobs*

Опис таблиці			
Поле	Опис	Тип даних	Ключ
Id	Ідентифікатор вакансії	int	Первинний
name	Назва вакансії	varchar	
date	Дата створення вакансії	int	
city_id	Ідентифікатор міста	int	Зовнішній
mail	Пошта вакансії	varchar	
phone	Телефон вакансії	varchar	
position	Посада вакансії	varchar	
salary	Заробітна плата	int	
employment_id	Ідентифікатор зайнятості	int	Зовнішній
rubric_id	Ідентифікатор рубрики	int	Зовнішній
description	Опис вакансії	text	
user_id	Ідентифікатор користувача	int	Зовнішній

Таблиця 2.2 – *Resume*

Опис таблиці			
Поле	Опис	Тип даних	Ключ
Id	Ідентифікатор резюме	int	Первинний
photo	Назва фото	varchar	
name	Ім'я користувача	int	
surname	Прізвище користувача	varchar	
date_birth	Дата народження	date	
city_id	Ідентифікатор міста	int	Зовнішній
phone	Номер телефону	varchar	

Продовження таблиці 2.2

mail	Пошта користувача	varchar	
position	Бажана посада	varchar	
salary	Заробітна плата	int	
employment_id	Ідентифікатор зайнятості	int	Зовнішній
rubric_id	Ідентифікатор рубрики	int	Зовнішній
description	Додаткова інформація	varchar	
user_id	Ідентифікатор користувача	int	Зовнішній

Таблиця 2.3 – Users

Опис таблиці			
Поле	Опис	Тип даних	Ключ
Id	Ідентифікатор користувача	int	Первинний
login	Логін користувача	varchar	
password	Пароль користувача	text	
mail	Пошта користувача	varchar	
type	Тип користувача	varchar	

### 2.3 Конструювання програмного забезпечення

Нижче наведено опис складових структур серверної частини:

Таблиця 2.4 – Опис класів системи

Клас/модуль	Опис
Database	Клас, що реалізує логіку роботу за базою даних.
Format	Клас, що містить логіку форматування та валідацію даних.
Session	Клас, що містить логіку роботи з сесіями користувача.

Змн.	Арк.	№ докум.	Підпис	Дата



Продовження таблиці 2.4

System	Клас, що містить логіку роботи з основними даними системи
User	Клас, який містить логіку обробки користувачів, збереження, авторизацію
Resume	Клас, який містить логіку обробки та створення резюме, обробку запитів на API
UserResume	Клас, який містить логіку обробки, редагування та видалення резюме
Vacancy	Клас, який містить логіку обробки та створення вакансій
UserVacancy	Клас, який містить логіку обробки, редагування та видалення вакансій

Таблиця 2.5 – Опис методів класів системи

Клас/Модуль	Метод	Опис
Database	connectDB ()	З'єднання з базою даних
Database	select(\$query)	Виконання запиту на вибірку даних з бази
Database	insert(\$query)	Виконання запиту на додавання даних до бази
Database	update (\$query)	Виконання запиту на оновлення даних в базі

## Продовження таблиці 2.5

Database	delete (\$query)	Виконання запиту на видалення даних з бази
Format	formatGetYear(\$date)	Вибірка року з повної дати
Format	formatDateToDB(\$date)	Форматування дати у вигляд для запису в базу даних
Format	formatDateNormal(\$date)	Форматування дати у звичний вигляд
Format	formatMonthYearNormal(\$date)	Вибірка місяця й року з повної дати
Format	formatGetDay(\$date)	Вибірка дня з повної дати
Format	formatGetMonth(\$date)	Вибірка місяця з повної дати
Format	validation(\$data)	Валідація даних
Session	init()	Початок нової сесії
Session	set(\$key, \$val)	Ініціалізація значення сесії за ключем
Session	get(\$key)	Вибірка значення сесії за ключем
Session	checkSession()	Переірка сесії користувача
Session	checkAdminSession()	Перевірка сесії адміністратора
Session	checkLogin()	Перевірка авторизації користувача

## Продовження таблиці 2.5

Session	destroy()	Видалення сесії
User	getUserData(\$data)	Авторизація користувача
User	userRegistration(\$data)	Реєстрація користувача
System	checkTypeUser(\$id)	Визначення типу користувача.
System	getCities()	Вибірка всіх міст з бази даних
System	getCityById(\$id)	Вибірка міста по ідентифікатору
System	addCity(\$data)	Додавання міста в базу даних
System	deleteCity(\$id)	Видалення міста з бази даних
System	getCourseById(\$id)	Вибірка курсу з бази даних по ідентифікатору
System	addCourse(\$data)	Додавання курсу в базу даних
System	getSkills()	Вибірка навичок з бази даних
System	getSkillsById(\$id)	Вибір навички з бази даних по ідентифікатору
System	addSkills (\$data)	Додати навичку до бази даних

## Продовження таблиці 2.5

System	deleteSkills(\$id)	Видалення навчки з бази даних
System	getCoursesSkills()	Вибірка курсів і навчок, які їм відповідають
System	getEducationLevel()	Вибірка рівнів освіти з бази даних
System	addEducationLevel(\$data)	Додати рівень освіти до бази даних
System	deleteEducationLevel(\$id)	Видалення рівня освіти з бази даних
System	getEmployments()	Вибірка типів зайнятості з бази даних
System	getEmploymentById(\$id)	Вибірка типів зайнятості з бази даних по ідентифікатору
System	addEmployment(\$data)	Додати тип зайнятості до бази даних
System	deleteEmployment(\$id)	Видалення типу зайнятості з бази даних
System	getSkillName(\$id)	Вибірка назви навчка з бази даних по ідентифікатору

## Продовження таблиці 2.5

System	getExperienceSkillName(\$id)	Вибірка рівня навичка з бази даних по ідентифікатору
System	addExperienceSkill(\$data)	Додати рівень навички до бази даних
System	deleteExperienceSkill(\$id)	Видалення рівня навички з бази даних
System	getLanguage()	Вибірка списку мов з бази даних
System	addLanguage(\$data)	Додати мову до бази даних
System	deleteLanguage(\$id)	Видалення мови з бази даних
System	getLanguageLevel()	Вибірка рівнів володіння мовою з бази даних
System	addLanguageLevel(\$data)	Додати рівень володіння мовою до бази даних
System	deleteLanguageLevel(\$id)	Видалення рівня володіння мовою з бази даних
System	getRubrics()	Вибірка списку рубрик з бази даних
System	getRubricById(\$id)	Вибірка рубрики з бази даних по ідентифікатору

## Продовження таблиці 2.5

System	addRubrics(\$data)	Додати рубрику до бази даних
System	deleteRubrics(\$id)	Видалення рубрики з бази даних
System	getScopes()	Вибірка списку сфер діяльності з бази даних
System	addScopes(\$data)	Додати сферу діяльності до бази даних
System	deleteScopes(\$id)	Видалення сфери діяльності з бази даних
System	getUniversity()	Вибірка списку університетів з бази даних
System	addUniversity(\$data)	Додати університет до бази даних
System	deleteUniversity(\$id)	Видалення університету з бази даних
System	findCoursesSkill(\$id)	Вибірка курсів з бази даних, які відповідають навичкам
System	findCoursesBySkill(\$id)	Формування масиву курсів і всіх навичків, які їм відповідають

## Продовження таблиці 2.5

System	getStatisticTable(\$table)	Вибірка статистичних даних з бази даних по таблиці
System	getStatisticUpdateTable(\$table)	Отримання результатів з АПІ по кількості вакансій по шуканому критерію
System	insertDataStatisticTable(\$arr, \$table)	Додавання отриманих даних з АПІ у базу даних
Resume	addResume(\$data)	Створити нове резюме
Resume	insertResume(\$name, \$surname, \$date_birth, \$city_id, \$phone, \$mail, \$position, \$salary, \$employment_id, \$rubric_id, \$description, \$user_id)	Додавання даних про резюме користувача в базу даних.
Resume	insertUserExperience(\$id, \$company, \$scope, \$old_position, \$start_job, \$end_job, \$achievements)	Додавання досвіду роботи користувача в базу даних.
Resume	insertUserEducation(\$id, \$education_level, \$university, \$faculty, \$specialty, \$start_university, \$end_university)	Додавання освіти користувача до бази даних
Resume	insertUserLanguageInfo(\$id, \$language, \$language_level)	Додавання мови користувача до бази даних

## Продовження таблиці 2.5

Resume	insertSkills(\$id, \$skills, \$experience_skill)	Додавання навичок користувача до бази даних
Resume	addPhotoToResume(\$id, \$photo)	Додавання фото до резюме
Resume	getCityById(\$id)	Вибірка міста з бази даних по ідентифікатору
Resume	getEmploymentById(\$id)	Вибірка типу зайнятості з бази даних по ідентифікатору
Resume	getUniversityById(\$id)	Вибірка університета з бази даних по ідентифікатору
Resume	getUserLanguageById(\$id)	Вибірка мови в резюме з бази даних по ідентифікатору
Resume	getUserLanguageLevelById(\$id)	Вибірка рівня мови в резюме з бази даних по ідентифікатору
Resume	getEducationLevelById(\$id)	Вибірка рівня освіти в резюме з бази даних по ідентифікатору
Resume	getVacancyAPI(\$keywords)	Отримання вкансій по ключовим словам за допомогою API



## Продовження таблиці 2.5

Resume	getPostsByTag(\$get_tag)	Отримання статей по тегам за допомогою API
UserResume	getAllUserResume(\$id)	Вибірка всіх резюме користувача
UserResume	getSkillResume(\$id)	Вибірка навичок в резюме користувача
UserResume	getSkillName(\$id)	Вибірка назв навичок з бази даних
UserResume	getExperienceSkillName(\$id)	Вибірка назви рівня навичок з бази даних
UserResume	getEducationLevelById(\$id)	Вибірка назви рівня освіти з бази даних по ідентифікатору
UserResume	getUniversityById(\$id)	Вибірка назви університету з бази даних по ідентифікатору
UserResume	getEducationResume(\$id)	Вибірка даних про освіту користувача в резюме з бази даних
UserResume	getScope(\$id)	Вибірка даних про сферу діяльності користувача в резюме з бази даних

## Продовження таблиці 2.5

UserResume	getExperienceResume(\$id)	Вибірка даних про досвід роботи користувача в резюме з бази даних
UserResume	getLanguageById(\$id)	Вибірка даних про мови користувача в резюме з бази даних
UserResume	getLanguageLevelById(\$id)	Вибірка даних про рівні мов користувача в резюме з бази даних
UserResume	getLanguageResume(\$id)	Вибірка даних про загальну інформацію про мови користувача в резюме з бази даних
UserResume	delDir(\$dir)	Видалення папки та всіх файлів в ній
UserResume	deleteResume(\$id)	Видалення резюме
UserResume	deleteUserExperience(\$id)	Видалення інформації про досвід роботи в резюме
UserResume	deleteUserEducation(\$id)	Видалення інформації про освіту в резюме
UserResume	deleteUserLanguageInfo(\$id)	Видалення інформації про знання мов в резюме
UserResume	deleteUserSkills(\$id)	Видалення інформації про навички в резюме

## Продовження таблиці 2.5

UserResume	getUserResumeById(\$id)	Вибірка даних про резюме користувача з бази даних по ідентифікатору
UserResume	getUserResumeSkill(\$id)	Вибірка даних про навички користувача в резюме з бази даних
UserResume	getUserResumeExperience(\$id)	Вибірка даних про досвід роботи користувача в резюме з бази даних
UserResume	getUserResumeEducation(\$id)	Вибірка даних про освіту користувача в резюме з бази даних
UserResume	getUserResumeLanguageInfo(\$id)	Вибірка даних про знання мов користувача в резюме з бази даних
UserResume	getCityById(\$id)	Вибірка даних про місто користувача в резюме з бази даних
UserResume	getEmploymentById(\$id)	Вибірка даних про тип зайнятості користувача в резюме з бази даних

## Продовження таблиці 2.5

UserResume	updateResume(\$name, \$surname, \$date_birth, \$city_id, \$phone, \$mail, \$position, \$salary, \$employment_id, \$rubric_id, \$description, \$id)	Оновлення даних в резюме
UserResume	updatePhotoResume(\$id, \$photo)	Оновлення фото в резюме
UserResume	updateUserExperience(\$id, \$company, \$scope, \$old_position, \$start_year_job, \$start_month_job, \$end_year_job, \$end_month_job, \$achievements)	Оновлення даних про досвід роботи користувача в резюме
UserResume	updateUserEducation(\$id, \$education_level, \$university, \$faculty, \$specialty, \$start_university, \$end_university)	Оновлення даних про освіту користувача в резюме
UserResume	updateUserLanguageInfo(\$id, \$language, \$language_level)	Оновлення даних про знання мов користувача в резюме
UserResume	updateSkills(\$id, \$skills, \$experience_skill)	Оновлення даних про навички користувача в резюме
UserResume	updateResumePDF(\$resume_id)	Оновлення даних pdf файлу про резюме користувача
UserResume	userUpdateResume(\$data)	Оновлення даних в резюме користувача

## Продовження таблиці 2.5

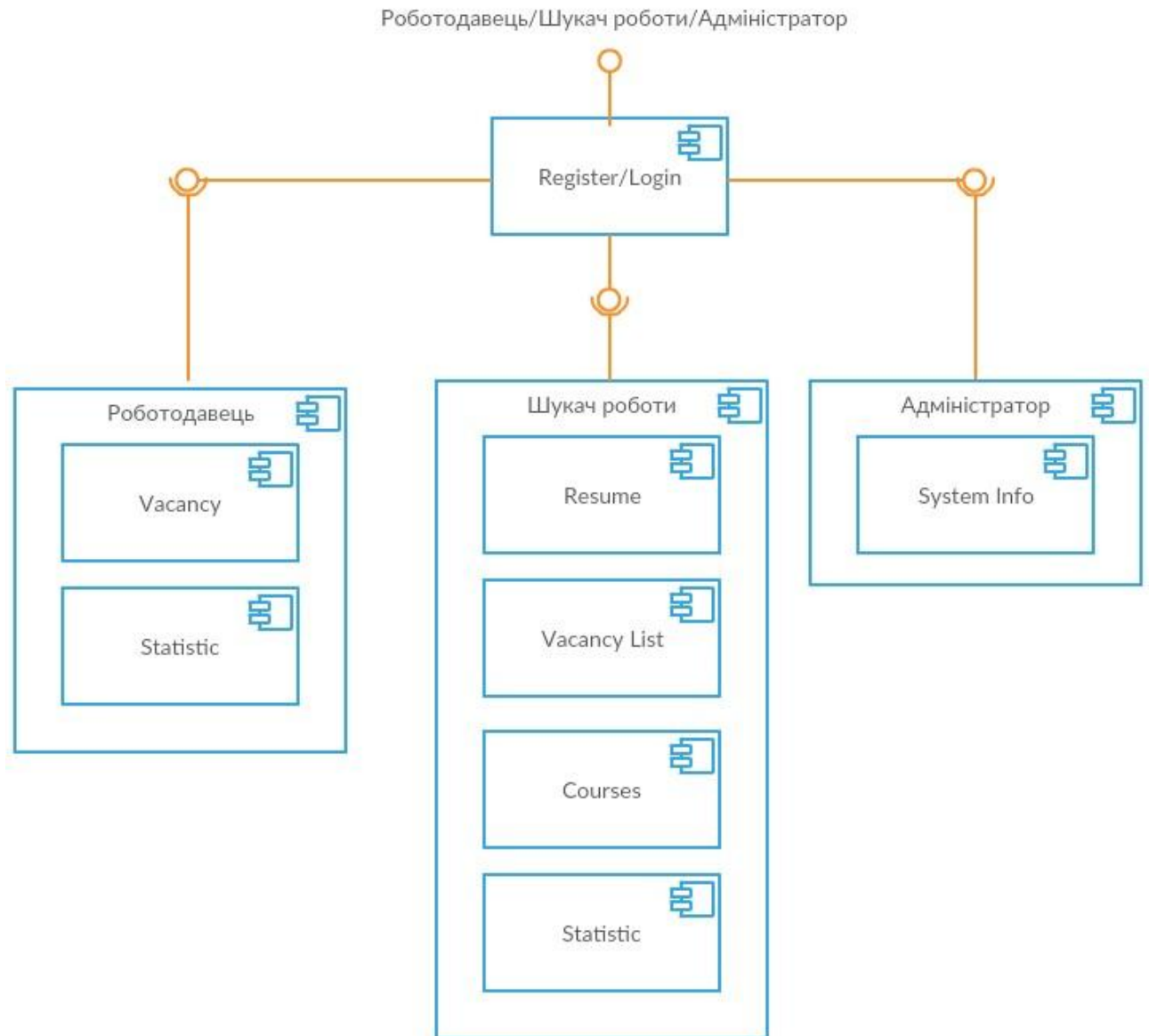
Vacancy	getVacancyId(\$id)	Вибірка даних про вакансію з бази даних по навичкам
Vacancy	getVacancyById(\$id)	Вибірка даних про вакансію з бази даних по ідентифікатору
Vacancy	insertVacancy(\$name, \$company, \$date, \$city_id, \$phone, \$mail, \$address, \$position, \$salary, \$employment_id, \$rubric_id, \$description, \$user_id)	Додавання даних про вакансію користувача в базу даних
Vacancy	addPhotoToVacancy(\$id, \$photo)	Додавання фото у вакансію користувача
Vacancy	insertVacancyEducation(\$id, \$education_level)	Додавання даних про освіту для вакансії в базу даних
Vacancy	insertVacancyLanguageInfo(\$id, \$language, \$language_level)	Додавання даних про знання мов для вакансії в базу даних
Vacancy	insertSkills(\$id, \$skills, \$experience_skill)	Додавання даних про навички для вакансії в базу даних
Vacancy	addVacancy(\$data)	Створити нову вакансію
UserVacancy	getAllUserVacancy(\$id)	Вибірка всіх вакансій користувача

## Продовження таблиці 2.5

UserVacancy	getUserVacancySkill(\$id)	Вибірка даних про навички для вакансії з бази даних
UserVacancy	getUserVacancyEducation(\$id)	Вибірка даних про освіту для вакансії з бази даних
UserVacancy	getUserVacancyLanguageInfo(\$id)	Вибірка даних про знання мов для вакансії з бази даних
UserVacancy	getSkillName(\$id)	Вибірка даних про назву навички для вакансії з бази даних
UserVacancy	getExperienceSkillName(\$id)	Вибірка даних про назву рівня навички для вакансії з бази даних
UserVacancy	getSkillVacancy(\$id)	Вибірка загальних даних про навички для вакансії
UserVacancy	getEducationVacancy(\$id)	Вибірка даних про освіту для вакансії з бази даних
UserVacancy	getEducationLevelById(\$id)	Вибірка даних про рівень освіти для вакансії з бази даних
UserVacancy	getLanguageVacancy(\$id)	Вибірка загальних даних про мову для вакансії з бази даних

## Продовження таблиці 2.5

UserVacancy	getLanguageById(\$id)	Вибірка даних про назву мови для вакансії з бази даних
UserVacancy	getLanguageLevelById(\$id)	Вибірка даних про рівень мови для вакансії з бази даних
UserVacancy	getPhotoVacancyById(\$id)	Вибірка фото вакансії з бази даних
UserVacancy	deleteVacancy(\$id)	Видалення вакансії
UserVacancy	updateVacancy(\$data)	Оновлення даних в вакансії
UserVacancy	updatePhotoVacancy(\$id, \$photo)	Оновлення фото вакансії
UserVacancy	updateVacancyEducation(\$id, \$education_level)	Оновлення даних про освіту в вакансії
UserVacancy	deleteVacancyEducation(\$id)	Видалення даних про освіту в вакансії
UserVacancy	updateVacancyLanguageInfo(\$id, \$language, \$language_level)	Оновлення даних про мову в вакансії
UserVacancy	deleteVacancyLanguageInfo(\$id)	Видалення даних про мову в вакансії
UserVacancy	updateSkills(\$id, \$skills, \$experience_skill)	Оновлення даних про навички в вакансії
UserVacancy	deleteVacancySkills(\$id)	Видалення даних про навички в вакансії
UserVacancy	userUpdateVacancy(\$data)	Оновлення даних в вакансії користувача



Для зберігання паролів було використано вбудовану PHP функцію `password_hash()`, яка створює хеш пароля використовуючи сильний, незворотний алгоритм хешування BCrypt, розроблений Нільсом Провосом і Девідом Мазьєра в 1999 році.

BCrypt заснований на криптоматичному алгоритмі блочного шифрування Blowfish і приймає форму адаптивної хеш-функції.

					КПІ.ІП-5114.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		86



Використовуючи ключовий фактор, BCrypt може регулювати вартість хешування. Зміни ключового фактора впливають на вихідні дані хеша. Таким чином, BCrypt залишається надзвичайно стійким до зломів, особливо до злому паролів, так званими «райдужними таблицями».

Цей ключовий фактор буде і надалі залишатися ключовою функцією, тому що це значно уповільнює швидкість хешування. В кінцевому підсумку уповільнення процесу злому, поки він не перестане бути життєздатною стратегією.

## 2.5 Висновки до розділу

У даному розділі були проілюстровані бізнес-процеси за допомогою діаграм BPMN, розроблено схему клієнт-серверної моделі, визначення архітектуру веб-додатку, обрано технології за якими буде розроблятися дипломний проект. Вибрано серверну та клієнтську мову програмування, обрано реляційну СУБД, було представлено структуру класів та функцій.

### 3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Аналіз якості ПЗ

Для того, щоб проаналізувати якість програмного забезпечення необхідно його протестувати. Тестування є одним з найголовніших етапів в розробці програмних продуктів. Тестування дозволяє завчасно виявити неправильну роботу програми, до того, як її почнуть масово використовувати, та вчасно виправити помилки при розробці. Адже на ранніх етапах легше виправити неточності в роботі програми, ніж коли вона вже використовується. Виправлення можуть зайняти декілька днів, і можуть вплинути на роботу інших модулів.

Стандарт ISO-9126 визначає шість характеристик програмного забезпечення:

- функціональність;
- надійність;
- зручність;
- ефективність;
- супроводжуваність;
- переносність.

Функціональність – це сукупність властивостей, які визначають спроможність програмного забезпечення виконувати в заданому середовищі перелік функцій відповідно вимогам до обробки і загальносистемним засобам.

Надійність – це множина атрибутів, котрі вказують на спроможність програмного забезпечення перетворювати вхідні дані на результати при умовах, що залежать від періоду часу (зношення й старіння програмного забезпечення не враховуються).

Зручність – це множина атрибутів, котрі вказують на необхідні або сприятливі умови для використання програмного забезпечення визначеним колом користувачів.

Ефективність – це множина атрибутів, які показують взаємозв'язок між рівнем виконання програмного забезпечення і кількістю використовуваних ресурсів у початкових умовах.

Супроводжуваність – це множина властивостей, які вказують на зусилля, котрі необхідно витратити на проведення модифікацій, таких як коригування, удосконалення й адаптацію програмного забезпечення зі зміною середовища, вимог або функціональних специфікацій.

Переносність – це множина показників, котрі вказують на здатність програмного забезпечення пристосовуватися до роботи при зміні середовища виконання.[5]

### 3.2 Опис процесів тестування

Будуть проводитись наступні функціональні типи тестів:

- тестування функціоналу збереження результатів у базу даних;
- тестування функціоналу створення PDF файлу з правильними результатами та можливість завантаження його.

Будуть проводитись наступні нефункціональні типи тестів:

- тестування роботи інтерфейсу користувача;
- тестування часу скачування PDF файлу;
- навантажувальне тестування сервера по обробці даних.

### 3.3 Опис контрольного прикладу

Назва: Введення даних

Опис: Авторизація неіснуючого користувача.

Кроки тестування:

- ввести в поле «Логін» неіснуючий логін;
- ввести в поле пароль дані;
- натиснути кнопку вхід.

Очікуваний результат: У поле записались введені данні, при натисненні кнопки «Вхід» виводиться інформація про помилку.

Реальний результат: У поле записались введені данні, при натисненні кнопки «Вхід» виводиться інформація про помилку.

Назва: Введення даних при створенні резюме

Опис: Перевірка реагування програми при незаповненому обов'язковому полі..

Кроки тестування:

- авторизуватися як «Пошукач»;
- перейти на сторінку створення резюме;
- заповнити всі обов'язкові поля окрім поля «І'мя»;
- натиснути кнопку «Зберегти».

Очікуваний результат: Після натиснення кнопки «Зберегти» виводиться інформація про необхідність заповнити обов'язкове поле.

Реальний результат: Після натиснення кнопки «Зберегти» виводиться інформація про необхідність заповнити обов'язкове поле.

Назва: Видалення резюме

Опис: Перевіряється видалення резюме з списку резюме.

Кроки тестування:

- авторизуватися як «Пошукач»;
- вибрати резюме, яке хочемо видалити;
- натиснути кнопку «Видалити».

Очікуваний результат: Після натиснення кнопки «Видалити» резюме зникає з списку резюме, і після перезагрузки сторінки не з'являється.

Реальний результат: Після натиснення кнопки «Видалити» резюме зникає з списку резюме, і після перезагрузки сторінки не з'являється.

### 3.4 Висновки до розділу

У даному розділі було проаналізовано якість програмного забезпечення, описані процеси тестування, та наведено три контрольних приклади тестування.

					КП.ІП-5114.045440.01.81	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		91

## 4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Розгортання програмного забезпечення

Для повного розгортання веб-додатку необхідно:

- перенести проект;
- перенести данні з бази даних.

#### 4.1.1 Перенесення проекту

Для перенесення проекту потрібно скопіювати всі файли і папки дипломного проекту в кореневу папку сервера.

#### 4.1.2 Перенесення даних з бази даних

Для перенесення даних з однієї бази даних в інше потрібно зробити експорт старої бази, і дані експортовані з старої бази імпортувати в нову.

### 4.2 Робота з програмним забезпеченням

Детальну інструкцію роботи з програмним забезпеченням наведено у документі “Керівництво користувача” та “Керівництво користувача(адміністратор)”.

### 4.3 Висновки до розділу

У даному розділі було описано процес впровадження та супроводу сервісу, зокрема розгортання програмного забезпечення та створено інструкцію користувача та адміністратора.

## ВИСНОВКИ

Під час виконання дипломного проекту був створений web-додаток для пошуку роботи, який дозволяє створювати, видаляти та змінювати резюме, шукати вакансії, створювати, видаляти та змінювати вакансії, шукати курси за обраним напрямом програмування, отримувати статистику по мовам програмування у вакансіях. Розроблена адміністративна панель, для адміністрування даних в системі.

Досліджено предметну область, визначено вимоги до комплексу задач, проаналізовано успішні рішення по напрямку теми дипломного проекту, розроблено функціональні і не функціональні вимоги, виконаний аналіз вимог до програмного забезпечення у розділі «Аналіз вимог до програмного забезпечення».

Змодельоване та проаналізоване програмне забезпечення, створена архітектура дипломного проекту, описано структуру бази даних, сконструйоване програмне забезпечення, проведено аналіз безпека даних в розділі «Моделювання та конструювання програмного забезпечення».

Проведений аналіз якості програмного забезпечення, описано процеси тестування дипломного проекту, описані контрольні приклади тестування в розділі «Аналіз якості та тестування програмного забезпечення».

Описано розгортання та перенесення програмного продукту, описано перенесення бази даних створено керівництво користувача та адміністратора, визначено роботу з програмним забезпеченням в розділі «Впровадження та супровід програмного забезпечення».

Розроблено необхідну проектну документацію, схему бази даних, схему варіантів використання та креслення екранних форм.

Створений web-додаток є логічно цілісним та коректним, надійно протестованим та готовим до використання.

					<b>КП.ІП-5114.045440.01.81</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		93

## ПЕРЕЛІК ПОСИЛАНЬ

- 1) Пошук роботи в Інтернеті, як не натрапити на аферистів [Електронний ресурс] – Режим доступу до ресурсу:  
<https://kha.dcz.gov.ua/publikaciya/poshuk-roboty-v-interneti-yak-ne-natrapyty-na-aferystiv>.
- 2) Корисна інформація - Працевлаштування [Електронний ресурс] – Режим доступу до ресурсу: <http://www.dut.edu.ua/ua/509-korisna-informaciya-pracevlashtuvannya>.
- 3) Що таке вакансія [Електронний ресурс] – Режим доступу до ресурсу:  
[http://otbetbl.com/Questions\\_companies/Chto\\_takoe\\_vakansiya\\_.html](http://otbetbl.com/Questions_companies/Chto_takoe_vakansiya_.html).
- 4) Сайти для пошуку роботи: який обрати? [Електронний ресурс] – Режим доступу до ресурсу: <https://careerfornewlife.com/blog/find/sayti-dlya-poshuku-roboti-yakiy-obrati/>.
- 5) АНАЛІЗ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ [Електронний ресурс] – Режим доступу до ресурсу:  
<http://oldconf.neasmo.org.ua/node/430>.
- 6) Що таке PHP? [Електронний ресурс] – Режим доступу до ресурсу:  
<http://phpist.com.ua/php/4-whatphp>.
- 7) Що таке API - Значення [Електронний ресурс] – Режим доступу до ресурсу: <https://sendpulse.ua/support/glossary/api-code>.
- 8) AJAX [Електронний ресурс] – Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/AJAX>.
- 9) JSON [Електронний ресурс] – Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/JSON>.
- 10) Bcrypt [Електронний ресурс] – Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/Bcrypt>.



**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматизованих систем обробки інформації і управління**

**“ЗАТВЕРДЖЕНО”**

В.о. завідувача кафедри

\_\_\_\_\_ О.А. Павлов

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**Web-застосування для пошуку роботи**

**Технічне завдання**

КПІ.ІП-5114.045440.03.91

**“ПОГОДЖЕНО”**

Керівник проекту:

\_\_\_\_\_ О.М. Клименко

Нормоконтроль:

\_\_\_\_\_ К.І. Ліщук

Виконавець:

\_\_\_\_\_ М.О. Миколаєнко

Київ – 2019 року

## ЗМІСТ

<b>1</b>	<b>НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ.....</b>	<b>3</b>
<b>2</b>	<b>ПІДСТАВА ДЛЯ РОЗРОБКИ.....</b>	<b>4</b>
<b>3</b>	<b>ПРИЗНАЧЕННЯ РОЗРОБКИ .....</b>	<b>5</b>
<b>4</b>	<b>ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>6</b>
4.1	Вимоги до функціональних характеристик.....	6
4.2	Вимоги до надійності .....	7
4.3	Умови експлуатації .....	8
4.4	Вимоги до складу і параметрів технічних засобів .....	8
4.5	Вимоги до інформаційної та програмної сумісності .....	8
4.6	Вимоги до маркування та пакування.....	9
4.7	Вимоги до транспортування та зберігання .....	9
<b>5</b>	<b>ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ .....</b>	<b>10</b>
<b>6</b>	<b>СТАДІЇ І ЕТАПИ РОЗРОБКИ.....</b>	<b>11</b>
<b>7</b>	<b>ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....</b>	<b>12</b>

## 1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

**Назва розробки:** Web-застосування для пошуку роботи

**Галузь застосування:**

Наведене технічне завдання поширюється на розробку Web-застосування, котре використовується для пошуку роботи і створення резюме, а також для пошуку курсів по мовам програмування, та призначена для студентів, яким важко знайти роботу самостійно, або створити якісне резюме.

					КПІ.ІП-5114.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

## 2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки Web-застосування для пошуку роботи є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації і управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім.Ігоря Сікорського).

					КПІ.ІП-5114.045440.03.91	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для студентів, які шукають роботу, які не можуть, або не знають, як зробити правильне резюме, а також для роботодавців, які готові брати студентів на роботу.

Метою розробки є підвищення якості резюме на ринку праці, спрощення пошуку роботи, зменшення часу на пошук роботи, підвищення зручності роботи у web-додатку при створенні резюме, створенні вакансії, або пошуку роботи.

					КПІ.ІП-5114.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

4.1.1 Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

#### 4.1.1.1 Для пошукача:

- реєстрація;
- авторизація;
- пошук вакансій;
- пошук статей по темам вказаних навичок в резюме;
- створення резюме;
- редагування резюме;
- видалення резюме;
- перегляд резюме;
- завантаження резюме в PDF форматі;
- пошук доступних курсів по заданим навичкам;
- перегляд статистичних діаграм за позиціями та мовами

програмування.

#### 4.1.1.2 Для роботодавця:

- реєстрація;
- авторизація;
- створення вакансій;
- редагування вакансій;
- видалення вакансій;
- перегляд вакансій;
- перегляд статистичних діаграм за позиціями та мовами

програмування.

					КП.ІП-5114.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

**4.1.1.3 Для адміністратора:**

- авторизація;
- додавання міст в базу даних;
- видалення міст з бази даних;
- додавання курсів в базу даних;
- видалення курсів з бази даних;
- додавання рівня освіти в базу даних;
- видалення рівня освіти з бази даних;
- додавання типу зайнятості в базу даних;
- видалення типу зайнятості з бази даних;
- додавання рівня навички в базу даних;
- видалення рівня навички з бази даних;
- додавання мови в базу даних;
- видалення мови з бази даних;
- додавання рівня мови в базу даних;
- видалення рівня мови з бази даних;
- додавання рубрик в базу даних;
- видалення рубрик з бази даних;
- додавання сфер діяльності в базу даних;
- видалення сфер діяльності з бази даних;
- додавання навичок в базу даних;
- видалення навичок з бази даних;
- додавання університетів в базу даних;
- видалення університетів з бази даних.

**4.1.2 Розробку виконати на платформі Windows 10****4.2 Вимоги до надійності**

- Передбачити контроль введення інформації;
- Передбачити захист від некоректних дій користувача;

					КП.ІП-5114.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

- Забезпечити цілісність інформації в базі даних;
- Організувати регулярне створення резервних копій бази даних.

#### 4.3 Умови експлуатації

4.3.1 Умови експлуатації згідно СанПін 2.2.2.542 – 96.

#### 4.3.2 Обслуговування

Вимоги до обслуговування не пред'являються.

#### 4.3.3 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не пред'являються.

#### 4.4 Вимоги до складу і параметрів технічних засобів

4.4.1 Мінімальна конфігурація технічних засобів:

а) конфігурація сервера:

- Apache – PHP – 7.2 x64;
- СУБД MySQL 5.6.

б) конфігурація клієнта:

- процесор 1.1 ГГц і вище, 2 ядра;
- 1 Гб оперативної пам'яті;
- Інтернет-з'єднання швидкістю не менше 2Мбіт/с.

#### 4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Програмне забезпечення повинно працювати під управлінням операційних систем сімейства Windows та підтримувати браузери Google Chrome, Opera, Microsoft Edge.

4.5.2 Вхідні дані повинні бути представлені в наступному форматі:

Якщо користувач авторизувався у якості шукача вакансій, вхідними даними для системи є:

					КПІ.ІП-5114.045440.03.91	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8



- особиста інформація про користувача, яку він заповнює в резюме;
- інформація в резюме про досвід роботи користувача;
- інформація в резюме про освіту користувача;
- інформація в резюме про знання мов користувача;
- додаткова інформація в резюме про користувача.

Якщо користувач авторизувався у якості роботодавця, вхідними даними для системи є:

- загальна інформація про вакансію;
- інформація про необхідну освіту в вакансії;
- інформація про необхідне знання мов в вакансії;
- додаткова інформація про вакансію.

#### 4.5.3 Результати повинні бути представлені в наступному форматі:

Якщо користувач авторизувався у якості шукача вакансій:

- список вакансій за вибраними навичками в резюме;
- список статей на теми навичок в резюме;
- резюме в PDF форматі.

Якщо користувач авторизувався у якості роботодавця:

- створена вакансія.

#### 4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

#### 4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

## 5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

5.2 Програмне забезпечення повинно мати довідникову систему

5.3 У склад супроводжувальної документації повинні входити наступні документи:

5.3.1 Пояснювальна записка не менше ніж на 70 аркушах формату А4 (без додатків 5.3.2 - 5.3.6).

5.3.2 Технічне завдання.

5.3.3 Керівництво користувача.

5.3.4 Програма та методика тестування

5.4 Графічна частина повинна бути виконана на аркушах формату А3, котрі включаються у якості додатків до пояснювальної записки:

5.4.1 Схема структурна варіантів використання

5.4.2 Схема бази даних

5.4.3 Креслення вигляду екранних форм

## 6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

	Назва етапу	Строк	Звітність
1	Вивчення літератури за тематикою проекту	01.04.2019	
2	Розробка технічного завдання	08.04.2019	Технічне завдання
3	Аналіз вимог та уточнення специфікацій	15.04.2019	Специфікації програмного забезпечення
4	Проектування структури програмного забезпечення, проектування компонентів	22.04.2019	Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму ...)
5	Програмна реалізація програмного забезпечення	29.04.2019	Тексти програмного забезпечення
6	Тестування програмного забезпечення	06.05.2019	Тести, результати тестування
7	Розробка матеріалів текстової частини проекту	13.05.2019	Пояснювальна записка.
8	Розробка матеріалів графічної частини проекту	20.05.2019	Графічний матеріал проекту
9	Оформлення технічної документації проекту	27.05.2019	Технічна документація

## 7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

### 7.1 Види випробувань

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІП-5114.045440.03.91	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

**ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ**

**Кафедра автоматизованих систем обробки інформації і управління**

«ЗАТВЕРДЖЕНО»

В.о. завідувача кафедри

\_\_\_\_\_ Павлов О.А.

(підпис)

(ініціали, прізвище)

“    ” \_\_\_\_\_ 2019 р.

**Web-застосування для пошуку роботи**

**Інструкція користувача**

КП.ІП-5114.045440.05.34

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ О.М. Клименко

Нормоконтроль:

\_\_\_\_\_ К.І. Ліщук

Виконав:

\_\_\_\_\_ М.О. Миколаєнко

Київ – 2019 року

## ЗМІСТ

<b>1</b>	<b>ЗАГАЛЬНІ ВІДОМОСТІ .....</b>	<b>3</b>
<b>2</b>	<b>ПОЧАТОК РОБОТИ .....</b>	<b>4</b>
2.1	Запуск застосунку .....	4
2.2	Підготовка застосунку до роботи .....	4
<b>3</b>	<b>РОБОТА ІЗ ЗАСТОСУНКОМ .....</b>	<b>7</b>
<b>4</b>	<b>РЕКОМЕНДАЦІЇ ПО ОСВОЄННЮ .....</b>	<b>28</b>

## 1 ЗАГАЛЬНІ ВІДОМОСТІ

Застосунок, який було створено призначений для студентів, які хочуть знайти роботу, та для роботодавців, які готові брати на роботу студентів.

Кожен користувач може реєструватись, авторизуватись, переглядати статистику по мовам програмування, та посадами у вакансіях.

Якщо користувач авторизувався, як пошукач, в нього є можливість створювати, редагувати, видаляти власні резюме, та шукати курси по обраній мові програмування.

Якщо користувач авторизувався, як роботодавець, в нього є можливість створювати, редагувати, видаляти власні вакансії.

## 2 ПОЧАТОК РОБОТИ

### 2.1 Запуск застосунку

Для початку роботи застосунку потрібно відкрити браузер і в адресну строку вписати адресу застосунку.

### 2.2 Підготовка застосунку до роботи

Перед початком роботи з застосунком необхідно зареєструватися, а після реєстрації авторизуватися в системі.

Після переходу на сайт застосунку буде відображена форма авторизації.

Логін

Пароль

Вхід

Реєстрація

Рисунок 2.1 – Форма авторизації в системі

Для того, щоб користувач авторизувався в системі, необхідно, щоб він був зареєстрований. Для цього необхідно перейти на сторінку реєстрації, натиснувши кнопку «Реєстрація».



Назад

Логін

Пароль

Повторіть пароль

Зареєструватись як

Пошукач ▼

Зареєструватись

Рисунок 2.2 – Форма реєстрації в системі

Після того, як користувач заповнить всі поля та вибере ким він реєструється (пошукач/роботодавець) і натисне кнопку «Зареєструватись», система переправить користувача на сторінку авторизації, на якій буде повідомлення про успішну реєстрацію.

Логін

Пароль

Вхід

Реєстрація

Реєстрація пройшла успішно

Рисунок 2.3 – Форма авторизації після реєстрації в системі

Після заповнення необхідних полів для авторизації користувачу відкриється особистий кабінет.

Логін

student

Пароль

.....|

Вхід

Реєстрація

Реєстрація пройшла успішно

Рисунок 2.4 – Форма авторизації з заповненими полями

В залежності від типу, який обрав користувач при реєстрації (пошукач/роботодавець), йому буде доступний різний функціонал, який дозволяє працювати з системою. Після авторизації користувачу відкриється його особистий кабінет.

### 3 РОБОТА ІЗ ЗАСТОСУНКОМ

Після авторизації користувача, як пошукача, йому відкриється особистий кабінет пошукача.

Ви зайшли як **student**

[Вихід](#) [Створити резюме](#) [Знайти курси](#) [Перегляд статистики](#)

У Вас поки що немає жодного резюме

#### Рисунок 3.1 – Особистий кабінет пошукача

Для створення нового резюме, користувачу потрібно натиснути кнопку «Створити резюме». Після цього йому відкриється сторінка створення резюме.

Ви зайшли як student

[Особистий кабінет](#)

Ваша фотографія (формати JPG, PNG):

 Файл не выбран

Ім'я\*

Прізвище\*

Дата народження\*

Місто\*

Номер телефону\*

E-mail\*

Бажана посада\*

Бажана заробітна плата\*

Вид зайнятості\*

Рубрика резюме\*

Навички\*

Рисунок 3.2 – Сторінка створення резюме

## Досвід роботи

X

Компанія

Сфера діяльності компанії

Посада

Початок роботи

Кінець роботи

Обов'язки та досягнення

[+Додати роботу](#)

Рисунок 3.3 – Сторінка створення резюме

## Освіта

X

Рівень освіти\*

Назва ВНЗ\*

Факультет\*

Спеціальність\*

Рік початку\*

Рік закінчення\*

[+Додати освіту](#)

## Володіння мовами

X

Мова

Рівень володіння

[+Додати мову](#)

Додаткова інформація про себе

Рисунок 3.4 – Сторінка створення резюме

Користувач має заповнити обов'язкові поля, які позначені зірочкою.  
Необов'язкові поля можна залишити порожніми.

Ви зайшли як student

Особистий кабінет

Ваша фотографія (формати JPG, PNG):

 Файл не выбран

Ім'я\*

Максим

Прізвище\*

Микопаєнко

Дата народження\*

29

Листопад

1997

Місто\*

Київ

Номер телефону\*

0932629767

E-mail\*

max.progeo@gmail.com

Бажана посада\*

PHP розробник

Бажана заробітна плата\*

25000

Вид зайнятості\*

Повна зайнятість

Рубрика резюме\*

IT

Навички\*

☒ php

1 - 2 роки

☒ html

3 - 5 років

☒ css

3 - 5 років

☒ JavaScript

1 - 2 роки

☐ java☐ python☐ C#☐ C++

Рисунок 3.5 – Заповнена форма створення резюме

Змн.	Арк.	№ докум.	Підпис	Дат

КПІ.ІП-5114.045440.05.34

Арк.

11

## Освіта

X

Рівень освіти\*

Вища ▼

Назва ВНЗ\*

КПІ ім. Ігоря Сікорського ▼

Факультет\*

Факультет інформаційних та обчислювальни

Спеціальність\*

Програмна інженерія

Рік початку\*

2015 ▼

Рік закінчення\*

2019 ▼

[+Додати освіту](#)

## Володіння мовами

X

Мова

українська ▼

Рівень володіння

Вільно ▼

[+Додати мову](#)

Додаткова інформація про себе

Додаткова інформація

**Зберегти**

Рисунок 3.6 – Заповнена форма створення резюме

Для створення резюме користувач повинен натиснути кнопку «Зберегти». Після цього користувачу відкриється сторінка зі створеним резюме, списком вакансій, з сайтів пошуку роботи, які можуть йому бути цікавими, списком цікавих статей за напрямком його навичок, та список вакансій з дипломного проекту, якщо існуючі вакансії будуть по напрямку його навичок.



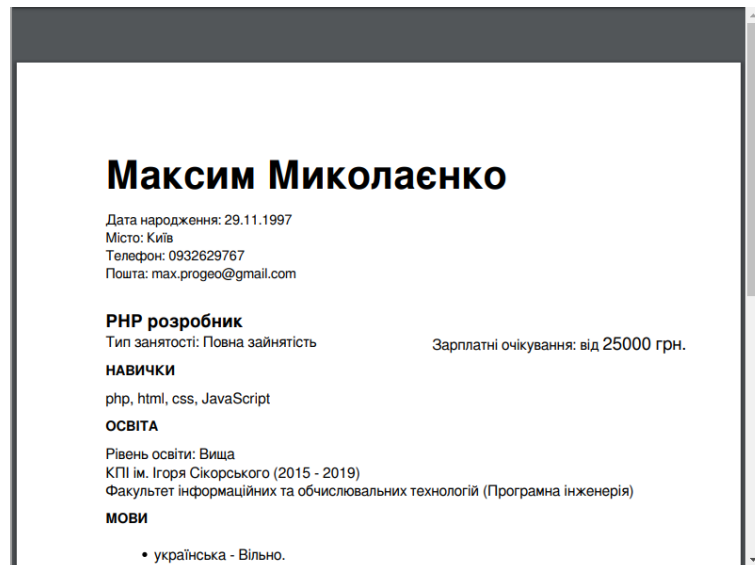
Ви заійшли як **student**[Особистий кабінет](#)

Рисунок 3.7 – Сторінка результату створення резюме

Вакансії, які можуть Вас зацікавити:

[Middle PHP & JavaScript Web Developer](#)

Місто: Київ

Заробітна плата: 1 200 - 2 000 \$

Зайнятість: Полная занятость

Опис: ...systems is "backend administration system". It is written in **PHP + JS, HTML and CSS** (it even has own client-server component system). The main... ..backend/frontend administrator application written in PHP5/**JavaScript** Required qualifications and skills: ~at least 1-2...

Джерело: jooble

Компанія: ShopIt Online Europe AB

25.04.2019

[PHP разработчик](#)

Місто: Київ

Заробітна плата: 15 000 грн.

Зайнятість: Полная занятость

Опис: Уровень ~Middle/Специалист Тип разработки ~Web development Требования: ~Опыт работы с **PHP/MySQL, HTML/CSS, Javascript** + jQuery, Ajax, Bootstrap, UI / UX, базовые понятия SEO, умение работать с JSON, XML ~Навык работы с FTP/phpMyAdmin/SSH/cPanel/AWS/WP Engine...

Джерело: jooble

Компанія: Смарт Медиа

06.06.2019

Рисунок 3.8 – Сторінка результату створення резюме

[PHP разработчик \(Битрикс\)](#)

Місто: Київ

Заробітна плата: 35 000 грн.

Зайнятість: Полная занятость

Опис: «Люксоптика» приглашает на работу **PHP** разработчика (Битрикс) Группа компаний ЛЮКСОПТИКА - это национальный лидер оптического... ..: ~Высшее образование; ~Отличное знание PHP, MySQL, (X)HTML, **CSS, JavaScript, AJAX**; ~Опыт работы с PHP-фреймворк Bitrix, работа с системой...

Джерело: jooble

Компанія: Люксоптика

11.05.2019

[Wordpress-розробник](#)

Місто: Київ

Заробітна плата: 800 \$

Зайнятість: Полная занятость

Опис: ...Ти повинен розуміти, про що мова: Знання MySQL/**JavaScript/JQuery/CSS/HTML**, REST API. Впевнені знання CMS WordPress; Вміння... ..документації; Досвід роботи з JSON, AJAX Базові знання: **PHP**, MySQL. Особисті якості: Відповідальність,...

Джерело: HeadHunter

Компанія: DL Agency

20.05.2019

Рисунок 3.9 – Сторінка результату створення резюме

					КПІ.ІП-5114.045440.05.34	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дат		

Цікаві статі по темам: php, html, css, JavaScript



Рисунок 3.10 – Сторінка результату створення резюме

Користувач може повернутися до особистого кабінету, де будуть відображатися всі його створені резюме.

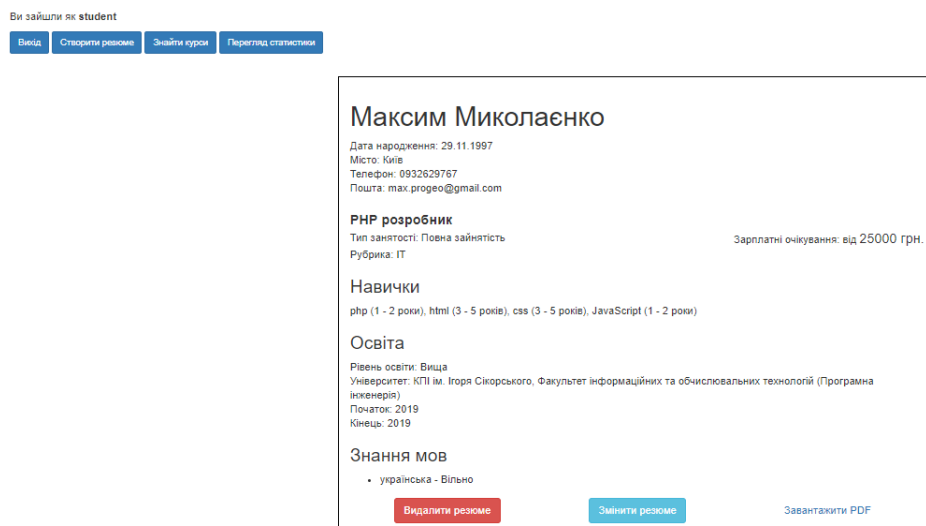


Рисунок 3.11 – Особистий кабінет користувача зі створеним резюме

Щоб змінити інформацію в резюме користувачу необхідно натиснути кнопку «Змінити резюме», після цього відкриється сторінка редагування резюме.

## Особистий кабінет

Ваша фотографія (формати JPG, PNG):

 Файл не выбран

Ім'я\*

Максим

Прізвище\*

Миколасенко

Дата народження\*

29 ▼

Листопад ▼

1997 ▼

Місто\*

Київ ▼

Номер телефону\*

0932629767

E-mail\*

max.progeo@gmail.com

Бажана посада\*

PHP розробник

Бажана заробітна  
плата\*

25000

Вид зайнятості\*

Повна зайнятість ▼

Рубрика резюме\*

ІТ ▼

Навички\*

☒ php

1 - 2 роки ▼

☒ html

3 - 5 років ▼

☒ css

3 - 5 років ▼

☒ JavaScript

1 - 2 роки ▼

☐ java☐ python☐ C#☐ C++

Рисунок 3.12 – Сторінка редагування резюме

## Освіта

X

Рівень освіти\*

Вища ▼

Назва ВНЗ\*

КПІ ім. Ігоря Сікорського ▼

Факультет\*

Факультет інформаційних

Спеціальність\*

Програмна інженерія

Рік початку\*

2015 ▼

Рік закінчення\*

2019 ▼

[+Додати освіту](#)

## Володіння мовами

X

Мова

українська ▼

Рівень володіння

Вільно ▼

[+Додати мову](#)Додаткова  
інформація про  
себе

Додаткова інформація

Зберегти

Рисунок 3.13 – Сторінка редагування резюме

Для зміни даних, необхідно редагувати потрібні поля.

## Освіта

✗

Рівень освіти\*

Вища ▼

Назва ВНЗ\*

КПІ ім. Ігоря Сікорця ▼

Факультет\*

Факультет інформаційн

Спеціальність\*

Програмна інженерія

Рік початку\*

2015 ▼

Рік закінчення\*

2019 ▼

[+Додати освіту](#)

## Володіння мовами

✗

Мова

українська ▼

Рівень володіння

Вільно ▼

[+Додати мову](#)Додаткова  
інформація про  
себе

Люблю навчатися.

**Зберегти**

Рисунок 3.14 – Зміна додаткової інформації

Після редагування всієї необхідної інформації потрібно натиснути кнопку «Зберегти». Після цього користувач повернеться до особистого кабінету, в якому буде змінено резюме.

Ви заійшли як student

Вийді

Створити резюме

Знайти курси

Перегляд статистики

## Максим Миколаєнко

Дата народження: 29.11.1997  
 Місто: Київ  
 Телефон: 0932629767  
 Пошта: max.progeo@gmail.com

## РНР розробник

Тип занятості: Повна зайнятість  
 Рубрика: IT

Зарплатні очікування: від 25000 грн.

## Навички

php (1 - 2 роки), html (3 - 5 років), css (3 - 5 років), JavaScript (1 - 2 роки)

## Освіта

Рівень освіти: Вища  
 Університет: КПІ ім. Ігоря Сікорського, Факультет інформаційних та обчислювальних технологій (Програма інженерія)  
 Початок: 2015  
 Кінець: 2019

## Знання мов

- українська - Вільно

## Додаткова інформація про себе

Люблю навчатися.

Видалити резюме

Змінити резюме

Завантажити PDF

Рисунок 3.15 – Резюме після зміни

Для того, щоб завантажити резюме в форматі PDF, користувач повинен натиснути кнопку «Завантажити PDF», після чого файл завантажиться на комп'ютер.

Ви заійшли як student

Вийді

Створити резюме

Знайти курси

Перегляд статистики

## Максим Миколаєнко

Дата народження: 29.11.1997  
 Місто: Київ  
 Телефон: 0932629767  
 Пошта: max.progeo@gmail.com

## РНР розробник

Тип занятості: Повна зайнятість  
 Рубрика: IT

Зарплатні очікування: від 25000 грн.

## Навички

php (1 - 2 роки), html (3 - 5 років), css (3 - 5 років), JavaScript (1 - 2 роки)

## Освіта

Рівень освіти: Вища  
 Університет: КПІ ім. Ігоря Сікорського, Факультет інформаційних та обчислювальних технологій (Програма інженерія)  
 Початок: 2015  
 Кінець: 2019

## Знання мов

- українська - Вільно

## Додаткова інформація про себе

Люблю навчатися.

Resume.pdf

Показати все

Рисунок 3.16 – Завантаження резюме на комп'ютер

Для видалення резюме, користувачу потрібно натиснути кнопку «Видалити резюме», після цього з'явиться спливаюче вікно з підтвердженням видалення.

Змн.	Арк.	№ докум.	Підпис	Дат

КПІ.ІП-5114.045440.05.34

Арк.

18

Ви зашли як student

Вийді Створити резюме Знайти курси Перегляд статистики

Підтвердіть дійство на странице діплом  
Видалити резюме?

OK Отмена

**Максим**

Дата народження: 29.11.1997  
Місто: Київ  
Телефон: 0932629767  
Пошта: max.progeo@gmail.com

**РНР розробник**  
Тип занятості: Повна зайнятість Зарплатні очікування: від 25000 грн.  
Рубрика: IT

**Навички**  
php (1 - 2 роки), html (3 - 5 років), css (3 - 5 років), JavaScript (1 - 2 роки)

**Освіта**  
Рівень освіти: Вища  
Університет: КПІ ім. Ігоря Сікорського, Факультет інформаційних та обчислювальних технологій (Програма інженерів)  
Початок: 2015  
Кінець: 2019

**Знання мов**  
• українська - Вільно

**Додаткова інформація про себе**  
Люблю навчатися.

Видалити резюме Змінити резюме Завантажити PDF

Рисунок 3.17 – Спливаюче вікно після натискання кнопки «Видалити резюме»

Після підтвердження видалення, резюме видаляється.

Ви зашли як student

Вийді Створити резюме Знайти курси Перегляд статистики

У Вас поки що немає жодного резюме

Рисунок 3.18 – Особистий кабінет після видалення резюме

Пошукач може знайти курси по мові програмування, яка його цікавить, для цього йому потрібно перебуваючи в особистому кабінеті натиснути кнопку «Знайти курси», після чого відкриється сторінка пошуку курсів.

Ви зашли як student

Особистий кабінет

Оберіть мову програмування

Знайти

Рисунок 3.19 – Сторінка пошуку курсів

Для того, щоб знайти курси, користувач повинен вибрати мову програмування з списку, та натиснути кнопку «Знайти». Після чого на сторінці з'явиться список курсів.

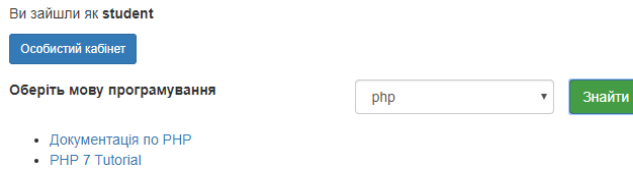


Рисунок 3.20 – Список курсів по обраній мові програмування

Після авторизації користувача, як роботодавця, йому відкриється особистий кабінет роботодавця.

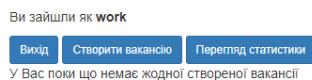


Рисунок 3.21 – Особистий кабінет роботодавця

Для створення нової вакансії, користувачу потрібно натиснути кнопку «Створити вакансію». Після цього йому відкриється сторінка створення вакансії.



Ви зайшли як work

[Особистий кабінет](#)

Ваша фотографія (формати JPG, PNG):

 Файл не выбран

Ваше ім'я\*

Компанія\*

Місто\*



Номер телефону\*

E-mail\*

Адреса\*

Посада\*

Заробітна плата\*

Вид зайнятості\*



Рубрика вакансії\*



Навички\*

Освіта

Рівень освіти\*



Рисунок 3.22 – Сторінка створення вакансії

## Володіння мовами

X

Мова\*

Рівень володіння\*

[+Додати мову](#)

Детальний опис вакансії\*

Детальний опис

Зберегти

Рисунок 3.23 – Сторінка створення вакансії

Користувач має заповнити обов'язкові поля, які позначені зірочкою.

Ви зайшли як work

Особистий кабінет

Ваша фотографія (формати JPG, PNG):

 Файл не вибран

Ваше ім'я\*

Максим

Компанія\*

Компанія

Місто\*

Київ

Номер телефону\*

0932629767

E-mail\*

max.progeo@gmail.com

Адреса\*

проспект Перемоги, 37

Посада\*

PHP розробник

Заробітна плата\*

25000

Вид зайнятості\*

Повна зайнятість

Рубрика вакансії\*

IT

Навички\*

☒ php

Менше року

☒ html

Менше року

☒ css

Менше року

☒ JavaScript

Менше року

☐ java☐ python☐ C#☐ C++

Рисунок 3.24 – Заповнена форма створення вакансії

Змн.	Арк.	№ докум.	Підпис	Дат

КПІ.ІП-5114.045440.05.34

Арк.

22

## Освіта

Рівень освіти\*

Вища

## Володіння мовами

X

Мова\*

англійська

Рівень володіння\*

Вище середнього

[+Додати мову](#)

Детальний опис вакансії\*

Супер вакансія.

Зберегти

Рисунок 3.25 – Заповнена форма створення вакансії

Для створення вакансії користувач повинен натиснути кнопку «Зберегти». Після цього користувачу відкриється особистий кабінет в якому буде створена вакансія.

Ви зашли як work

Вийти

Створити вакансію

Перегляд статистики

## PHP розробник

Заробітна плата: від 25000 грн.

Компанія: Компанія

Місто: Київ

Адреса: проспект Перемоги, 37

Телефон: 0932629767

Пошта: max.pirogo@gmail.com

Тип занятості: Повна зайнятість

Рубрика: IT

## Навички

php (Менше року), html (Менше року), css (Менше року), JavaScript (Менше року)

## Освіта

Рівень освіти: Вища

## Знання мов

• англійська - Вище середнього

## Додаткова інформація про вакансію

Супер вакансія.

Дата публікації: 06.06.2019

Видалити вакансію

Змінити вакансію

Рисунок 3.26 – Особистий кабінет зі створеною вакансією

					КПІ.ІП-5114.045440.05.34	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дат		

Щоб змінити інформацію в вакансії користувачу необхідно натиснути кнопку «Змінити вакансію», після цього відкриється сторінка редагування вакансії.

**Особистий кабінет**

Ваша фотографія (формати JPG, PNG):  
 Файл не вибран

Ваше ім'я\*

Компанія\*

Місто\*

Номер телефону\*

E-mail\*

Адреса\*

Бажана посада\*

Бажана заробітна плата\*

Вид зайнятості\*

Рубрика вакансії\*

Навички\*

<input checked="" type="checkbox"/> php	<input type="text" value="Менше року"/>	<input checked="" type="checkbox"/> html	<input type="text" value="Менше року"/>	<input checked="" type="checkbox"/> css	<input type="text" value="Менше року"/>	<input checked="" type="checkbox"/> JavaScript	<input type="text" value="Менше року"/>
<input type="checkbox"/> java	<input type="text"/>	<input type="checkbox"/> python	<input type="text"/>	<input type="checkbox"/> C#	<input type="text"/>	<input type="checkbox"/> C++	<input type="text"/>

Рисунок 3.27 – Сторінка редагування вакансії

## Освіта

X

Рівень освіти

[Додати освіту](#)

## Володіння мовами

X

Мова

Рівень володіння

[Додати мову](#)Додаткова  
інформація про  
себе

Рисунок 3.28 – Сторінка редагування вакансії

					КПІ.ІП-5114.045440.05.34	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		24

Для зміни даних, необхідно редагувати потрібні поля.

### Освіта

X

Рівень освіти

Незакінчена вища ▼

[Додати освіту](#)

### Володіння мовами

X

Мова

англійська ▼

Рівень володіння

Вище середнього ▼

[Додати мову](#)Додаткова  
інформація про  
себе

Супер вакансія.

**Зберегти**

Рисунок 3.29 – Зміна рівня освіти

Після всіх необхідних правок інформації потрібно натиснути кнопку «Зберегти». Після цього користувач повернеться до особистого кабінету, в якому буде змінена вакансія.

Змн.	Арк.	№ докум.	Підпис	Дат

Ви зашли як work

Вийді

Створити вакансію

Перегляд статистики

## PHP розробник

Заробітна плата: від 25000 грн.

Компанія: Компанія

Місто: Київ

Адреса: проспект Перемоги, 37

Телефон: 0932629767

Пошта: max.progeo@gmail.com

Тип занятості: Повна зайнятість

Рубрика: IT

## Навички

php (Менше року), html (Менше року), css (Менше року), JavaScript (Менше року)

## Освіта

Рівень освіти: Незакінчена вища

## Знання мов

- англійська - Вище середнього

## Додаткова інформація про вакансію

Супер вакансія.

Дата публікації: 06.06.2019

Видалити вакансію

Змінити вакансію

## Рисунок 3.30 – Вакансія після зміни

Для видалення вакансії, користувачу потрібно натиснути кнопку «Видалити вакансію», після цього з'явиться спливаюче вікно з підтвердженням видалення.

Ви зашли як work

Вийді

Створити вакансію

Перегляд статистики

Потвердите действие на странице diptom

Видалити вакансію?

ОК

Отмена

## PHP розробник

Заробітна плата: від 25000 грн.

Компанія: Компанія

Місто: Київ

Адреса: проспект Перемоги, 37

Телефон: 0932629767

Пошта: max.progeo@gmail.com

Тип занятості: Повна зайнятість

Рубрика: IT

## Навички

php (Менше року), html (Менше року), css (Менше року), JavaScript (Менше року)

## Освіта

Рівень освіти: Незакінчена вища

## Знання мов

- англійська - Вище середнього

## Додаткова інформація про вакансію

Супер вакансія.

Дата публікації: 06.06.2019

Видалити вакансію

Змінити вакансію

## Рисунок 3.31 – Спливаюче вікно після натискання кнопки «Видалити вакансію»

Після підтвердження видалення, вакансія видаляється.

					КПІ.ІП-5114.045440.05.34	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дат		

Ви зайшли як work

Вихід

Створити вакансію

Перегляд статистики

У Вас поки що немає жодної створеної вакансії

## Рисунок 3.32 – Особистий кабінет після видалення вакансії

Для того, щоб переглянути статистику, користувач повинен перейти на сторінку «Перегляд статистики».

Ви зайшли як work

Особистий кабінет

Оберіть критерій

Показати статистику

Оновити статистику

## Рисунок 3.33 – Сторінка перегляду статистики

На сторінці статистики необхідно обрати критерій, за яким користувач хоче, подивитись статистику, і натиснути кнопку «Показати статистику».

Ви зайшли як work

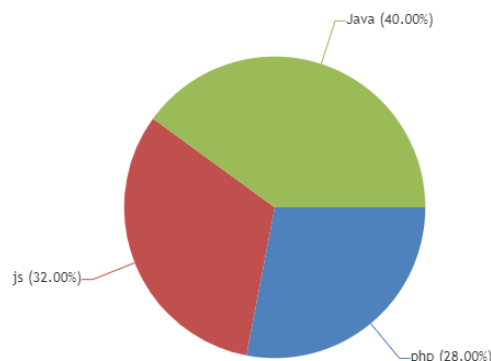
Особистий кабінет

Оберіть критерій

Показати статистику

Оновити статистику

Статистика: По мовам програмування



## Рисунок 3.34 – Показ статистики по мовам програмування

Для оновлення статистики необхідно натиснути кнопку «Оновлення статистики».

#### 4 РЕКОМЕНДАЦІЇ ПО ОСВОЄННЮ

Для кращого розуміння роботи із застосунком, користувач повинен мати навички роботи з веб-сервісами, а також бути ознайомленим з даною інструкцією.

					КПІ.ІП-5114.045440.05.34	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дат		



**ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ**  
**Кафедра автоматизованих систем обробки інформації і управління**

«ЗАТВЕРДЖЕНО»

В.о. завідувача кафедри

\_\_\_\_\_ Павлов О.А.  
(підпис) (ініціали, прізвище)

“ ” \_\_\_\_\_ 2019 р.

**Web- застосування для пошуку роботи**

**Інструкція адміністратора**

КП.ІП-5114.045440.06.34

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ О.М. Клименко

Нормоконтроль:

\_\_\_\_\_ К.І. Ліщук

Виконав:

\_\_\_\_\_ М.О. Миколаєнко

Київ – 2019 року

## ЗМІСТ

<b>1</b>	<b>ЗАГАЛЬНІ ВІДОМОСТІ .....</b>	<b>3</b>
<b>2</b>	<b>ПОЧАТОК РОБОТИ .....</b>	<b>4</b>
2.1	Запуск застосунку .....	4
2.2	Підготовка застосунку до роботи .....	4
<b>3</b>	<b>РОБОТА ІЗ ЗАСТОСУНКОМ .....</b>	<b>6</b>
<b>4</b>	<b>РЕКОМЕНДАЦІЇ ПО ОСВОЄННЮ .....</b>	<b>21</b>

## 1 ЗАГАЛЬНІ ВІДОМОСТІ

Застосунок, який було створено призначений для студентів, які хочуть знайти роботу, та для роботодавців, які готові брати на роботу студентів.

Адміністратор має можливість авторизації в системі та редагування даних, які пропонуються користувачам обирати, при роботі з веб-застосунком.

					КПІ.ІП-5114.045440.06.34	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дат		

## 2 ПОЧАТОК РОБОТИ

### 2.1 Запуск застосунку

Для початку роботи застосунку потрібно відкрити браузер і в адресну строку вписати адресу застосунку.

### 2.2 Підготовка застосунку до роботи

Перед початком роботи з застосунком необхідно зареєструватися, а після реєстрації авторизуватися в системі.

Після переходу на сайт застосунку буде відображена форма авторизації.

Логін

Пароль

Вхід

Реєстрація

Рисунок 2.1 – Форма авторизації в системі

Для того, щоб адміністратор увійшов в свій особистий кабінет, йому необхідно авторизуватися в системі.

Змн.	Арк.	№ докум.	Підпис	Дат

Логін

admin

Пароль

.....

Вхід

Реєстрація

Рисунок 2.2 – Авторизація адміністратора в системі

### 3 РОБОТА ІЗ ЗАСТОСУНКОМ

Після авторизації адміністратору відкриється його особистий кабінет.

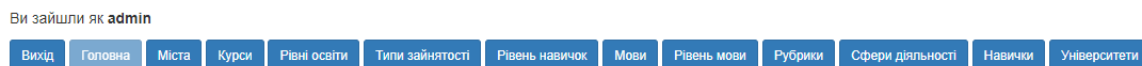


Рисунок 3.1 – Особистий кабінет адміністратора

Для того, щоб відредагувати дані, які користувач може обирати, для роботи з веб-застосунком, адміністратору необхідно перейти на відповідну сторінку. Щоб змінити інформацію про міста, потрібно перейти на сторінку «Міста».

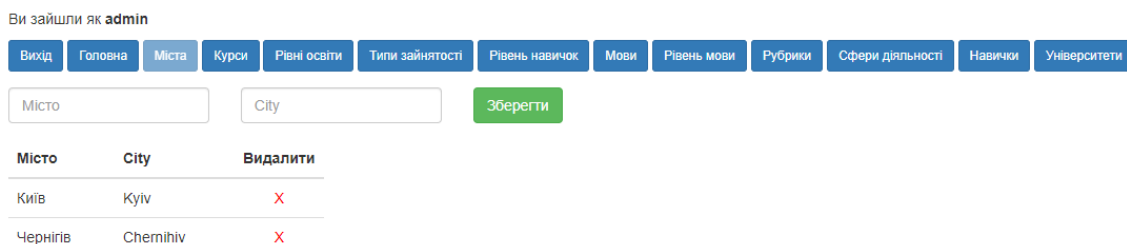


Рисунок 3.2 – Сторінка інформації про міста

Щоб, додати нове місто, адміністратор повинен вписати назву міста українською і англійською у відповідні поля.

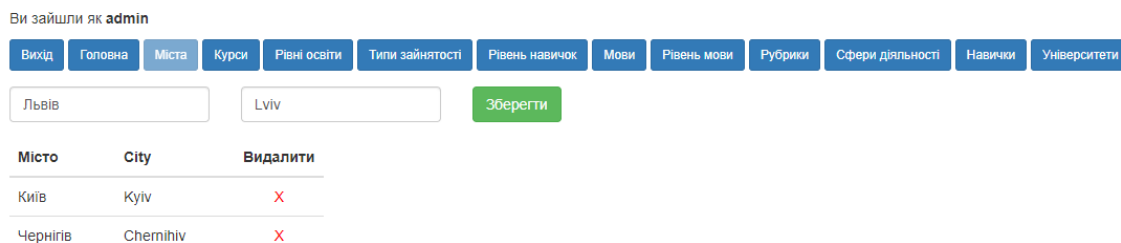


Рисунок 3.3 – Заповнена форма додавання міста

Для додавання нового міста, адміністратор повинен натиснути кнопку «Зберегти».

Ви заійшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Місто	City	Видалити
Київ	Kyiv	X
Чернігів	Chernihiv	X
Львів	Lviv	X

Рисунок 3.4 – Результат додавання нового міста

Щоб видалити місто необхідно натиснути на червоний хрестик біля назви міста, яке адміністратор хоче видалити.

Ви заійшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Місто	City	Видалити
Київ	Kyiv	X
Чернігів	Chernihiv	X

Рисунок 3.5 – Результат видалення міста

Щоб змінити інформацію про курси, потрібно перейти на сторінку «Курси».

Ви заійшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

php  
 html  
 css  
 JavaScript

Назва	Посилання	Навички	Видалити
Документація по PHP	<a href="https://www.php.net/manual/ru/">https://www.php.net/manual/ru/</a>	php	X
PHP 7 Tutorial	<a href="https://www.w3schools.com/php7/">https://www.w3schools.com/php7/</a>	php	X

Рисунок 3.6 – Сторінка інформації про курси

Щоб, додати новий курс, адміністратор повинен вписати назву курсу і посилання на нього у відповідні поля, та обрати з списку навички, які можна здобути завдяки курсу.

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)



Назва	Посилання	Навички	Видалити
Документація по PHP	<a href="https://www.php.net/manual/ru/">https://www.php.net/manual/ru/</a>	php	<input type="button" value="X"/>
PHP 7 Tutorial	<a href="https://www.w3schools.com/php7/">https://www.w3schools.com/php7/</a>	php	<input type="button" value="X"/>

Рисунок 3.7 – Заповнена форма додавання курсу

Для додавання нового курсу, адміністратор повинен натиснути кнопку «Зберегти».

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)



Назва	Посилання	Навички	Видалити
Документація по PHP	<a href="https://www.php.net/manual/ru/">https://www.php.net/manual/ru/</a>	php	<input type="button" value="X"/>
PHP 7 Tutorial	<a href="https://www.w3schools.com/php7/">https://www.w3schools.com/php7/</a>	php	<input type="button" value="X"/>
PHP Academy	<a href="https://php-academy.kiev.ua/">https://php-academy.kiev.ua/</a>	php	<input type="button" value="X"/>

Рисунок 3.8 – Результат додавання нового курсу

Щоб видалити курс необхідно натиснути на червоний хрестик біля назви курсу, яке адміністратор хоче видалити.

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)



Назва	Посилання	Навички	Видалити
Документація по PHP	<a href="https://www.php.net/manual/ru/">https://www.php.net/manual/ru/</a>	php	<input type="button" value="X"/>
PHP 7 Tutorial	<a href="https://www.w3schools.com/php7/">https://www.w3schools.com/php7/</a>	php	<input type="button" value="X"/>

Рисунок 3.9 – Результат видалення курсу

Щоб змінити інформацію про рівень освіти, потрібно перейти на сторінку «Рівні освіти».



Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рівень	Видалити
Вища	<input type="checkbox"/>
Незакінчена вища	<input type="checkbox"/>
Середньо-спеціальна	<input type="checkbox"/>
Середня	<input type="checkbox"/>

Рисунок 3.10 – Сторінка інформації про рівень освіти

Щоб, додати новий рівень освіти, адміністратор повинен вписати рівень освіти у поле вводу.

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рівень	Видалити
Вища	<input type="checkbox"/>
Незакінчена вища	<input type="checkbox"/>
Середньо-спеціальна	<input type="checkbox"/>
Середня	<input type="checkbox"/>
Науковий	<input type="checkbox"/>

Рисунок 3.11 – Заповнена форма додавання рівня освіти

Для додавання нового рівня освіти, адміністратор повинен натиснути кнопку «Зберегти».

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рівень	Видалити
Вища	<input type="checkbox"/>
Незакінчена вища	<input type="checkbox"/>
Середньо-спеціальна	<input type="checkbox"/>
Середня	<input type="checkbox"/>
Науковий	<input type="checkbox"/>

Рисунок 3.12 – Результат додавання нового рівня освіти

Щоб видалити рівень освіти необхідно натиснути на червоний хрестик біля рівня освіти, яке адміністратор хоче видалити.

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рівень	Видалити
Вища	<input type="checkbox"/>
Незакінчена вища	<input type="checkbox"/>
Середньо-спеціальна	<input type="checkbox"/>
Середня	<input type="checkbox"/>

Рисунок 3.13 – Результат видалення рівня освіти

Щоб змінити інформацію про типи зайнятості, потрібно перейти на сторінку «Типи зайнятості».

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Тип зайнятості	Видалити
Повна зайнятість	<input type="checkbox"/>
Неповна зайнятість	<input type="checkbox"/>
Віддалена робота	<input type="checkbox"/>
Стажировка	<input type="checkbox"/>

Рисунок 3.14 – Сторінка інформації про типи зайнятості

Щоб додати новий тип зайнятості, адміністратор повинен вписати тип зайнятості у поле вводу.

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Тип зайнятості	Видалити
Повна зайнятість	<input type="checkbox"/>
Неповна зайнятість	<input type="checkbox"/>
Віддалена робота	<input type="checkbox"/>
Стажировка	<input type="checkbox"/>

Рисунок 3.15 – Заповнена форма додавання типу зайнятості

Для додавання нового типу зайнятості, адміністратор повинен натиснути кнопку «Зберегти».

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Тип зайнятості

Зберегти

Тип зайнятості	Видалити
Повна зайнятість	X
Неповна зайнятість	X
Віддалена робота	X
Стажировка	X
Практика	X

Рисунок 3.16 – Результат додавання типу зайнятості

Щоб видалити тип зайнятості необхідно натиснути на червоний хрестик біля типу зайнятості, яке адміністратор хоче видалити.

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Тип зайнятості

Зберегти

Тип зайнятості	Видалити
Повна зайнятість	X
Неповна зайнятість	X
Віддалена робота	X
Стажировка	X

Рисунок 3.17 – Результат видалення типу зайнятості

Щоб змінити інформацію про рівень навичок, потрібно перейти на сторінку «Рівень навичок».

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рівень навичка

Зберегти

Рівень	Видалити
Без досвіду	X
Менше року	X
1 - 2 роки	X
3 - 5 років	X
Більше 5 років	X

Рисунок 3.18 – Сторінка інформації про рівень навичок

Щоб, додати новий рівень навичок, адміністратор повинен вписати рівень навичку у поле вводу.

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Більше 10 років

Зберегти

Рівень	Видалити
Без досвіду	<a href="#">×</a>
Менше року	<a href="#">×</a>
1 - 2 роки	<a href="#">×</a>
3 - 5 років	<a href="#">×</a>
Більше 5 років	<a href="#">×</a>

Рисунок 3.19 – Заповнена форма додавання рівня навичок

Для додавання нового рівня навичку, адміністратор повинен натиснути кнопку «Зберегти».

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рівень навичка

Зберегти

Рівень	Видалити
Без досвіду	<a href="#">×</a>
Менше року	<a href="#">×</a>
1 - 2 роки	<a href="#">×</a>
3 - 5 років	<a href="#">×</a>
Більше 5 років	<a href="#">×</a>
Більше 10 років	<a href="#">×</a>

Рисунок 3.20 – Результат додавання нового рівня навичку

Щоб видалити рівень навичку необхідно натиснути на червоний хрестик біля рівня навичку, який адміністратор хоче видалити.

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рівень навичка

Зберегти

Рівень	Видалити
Без досвіду	<a href="#">×</a>
Менше року	<a href="#">×</a>
1 - 2 роки	<a href="#">×</a>
3 - 5 років	<a href="#">×</a>
Більше 5 років	<a href="#">×</a>

Рисунок 3.21 – Результат видалення рівня навичку

Щоб змінити інформацію про мови, потрібно перейти на сторінку «Мови».

Ви заходите як **admin**
[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)
[Зберегти](#)

Мова	Видалити
українська	<a href="#">×</a>
російська	<a href="#">×</a>
англійська	<a href="#">×</a>
німецька	<a href="#">×</a>

Рисунок 3.22 – Сторінка інформації про мови

Щоб, додати нову мову, адміністратор повинен вписати мову у поле вводу.

Ви заходите як **admin**
[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)
[Зберегти](#)

Мова	Видалити
українська	<a href="#">×</a>
російська	<a href="#">×</a>
англійська	<a href="#">×</a>
німецька	<a href="#">×</a>

Рисунок 3.23 – Заповнена форма додавання мови

Для додавання нової мови, адміністратор повинен натиснути кнопку «Зберегти».

Ви заходите як **admin**
[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)
[Зберегти](#)

Мова	Видалити
українська	<a href="#">×</a>
російська	<a href="#">×</a>
англійська	<a href="#">×</a>
німецька	<a href="#">×</a>
іспанська	<a href="#">×</a>

Рисунок 3.24 – Результат додавання нової мови

Щоб видалити мову необхідно натиснути на червоний хрестик біля назви мови, яку адміністратор хоче видалити.

Ви зашли як **admin**
[Вихід](#) [Головна](#) [Міста](#) [Курси](#) [Рівні освіти](#) [Типи зайнятості](#) [Рівень навичок](#) [Мови](#) [Рівень мови](#) [Рубрики](#) [Сфери діяльності](#) [Навички](#) [Університети](#)

Мова	Видалити
українська	<input type="checkbox"/>
російська	<input type="checkbox"/>
англійська	<input type="checkbox"/>
німецька	<input type="checkbox"/>

Рисунок 3.25 – Результат видалення мови

Щоб змінити інформацію про рівні мови, потрібно перейти на сторінку «Рівень мови».

Ви зашли як **admin**
[Вихід](#) [Головна](#) [Міста](#) [Курси](#) [Рівні освіти](#) [Типи зайнятості](#) [Рівень навичок](#) [Мови](#) [Рівень мови](#) [Рубрики](#) [Сфери діяльності](#) [Навички](#) [Університети](#)

Рівень мови	Видалити
Базовий	<input type="checkbox"/>
Нижче середнього	<input type="checkbox"/>
Середній	<input type="checkbox"/>
Вище середнього	<input type="checkbox"/>
Поглиблений	<input type="checkbox"/>
Вільно	<input type="checkbox"/>

Рисунок 3.26 – Сторінка інформації про рівні мови

Щоб, додати новий рівень мови, адміністратор повинен вписати назву рівня мови у поле вводу.

Ви зашли як **admin**
[Вихід](#) [Головна](#) [Міста](#) [Курси](#) [Рівні освіти](#) [Типи зайнятості](#) [Рівень навичок](#) [Мови](#) [Рівень мови](#) [Рубрики](#) [Сфери діяльності](#) [Навички](#) [Університети](#)

Рівень мови	Видалити
Базовий	<input type="checkbox"/>
Нижче середнього	<input type="checkbox"/>
Середній	<input type="checkbox"/>
Вище середнього	<input type="checkbox"/>
Поглиблений	<input type="checkbox"/>
Вільно	<input type="checkbox"/>

Рисунок 3.27 – Заповнена форма додавання рівня мови

Для додавання нового рівня мови, адміністратор повинен натиснути кнопку «Зберегти».

Ви зашли як **admin**
[Вихід](#) [Головна](#) [Міста](#) [Курси](#) [Рівні освіти](#) [Типи зайнятості](#) [Рівень навичок](#) [Мови](#) [Рівень мови](#) [Рубрики](#) [Сфери діяльності](#) [Навички](#) [Університети](#)

Рівень мови

Зберегти

Рівень мови	Видалити
Базовий	<a href="#">×</a>
Нижче середнього	<a href="#">×</a>
Середній	<a href="#">×</a>
Вище середнього	<a href="#">×</a>
Поглиблений	<a href="#">×</a>
Вільно	<a href="#">×</a>
Рідна	<a href="#">×</a>

Рисунок 3.28 – Результат додавання нового рівня мови

Щоб видалити рівень мови необхідно натиснути на червоний хрестик біля рівня мови, який адміністратор хоче видалити.

Ви зашли як **admin**
[Вихід](#) [Головна](#) [Міста](#) [Курси](#) [Рівні освіти](#) [Типи зайнятості](#) [Рівень навичок](#) [Мови](#) [Рівень мови](#) [Рубрики](#) [Сфери діяльності](#) [Навички](#) [Університети](#)

Рівень мови

Зберегти

Рівень мови	Видалити
Базовий	<a href="#">×</a>
Нижче середнього	<a href="#">×</a>
Середній	<a href="#">×</a>
Вище середнього	<a href="#">×</a>
Поглиблений	<a href="#">×</a>
Вільно	<a href="#">×</a>

Рисунок 3.29 – Результат видалення рівня мови

Щоб змінити інформацію про рубрики, потрібно перейти на сторінку «Рубрики».

Ви зашли як **admin**
[Вихід](#) [Головна](#) [Міста](#) [Курси](#) [Рівні освіти](#) [Типи зайнятості](#) [Рівень навичок](#) [Мови](#) [Рівень мови](#) [Рубрики](#) [Сфери діяльності](#) [Навички](#) [Університети](#)

Рубрика

Зберегти

Рубрики	Видалити
ІТ	<a href="#">×</a>
Маркетинг - Реклама - PR	<a href="#">×</a>
HR спеціалісти - Бізнес-тренери	<a href="#">×</a>
Наука - Освіта - Переклад	<a href="#">×</a>

Рисунок 3.30 – Сторінка інформації про рубрики

Щоб, додати нову рубрику, адміністратор повинен вписати назву рубрики у відповідне поле.

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Дизайн - Графіка - Фото

Зберегти

Рубрики	Видалити
ІТ	<a href="#">×</a>
Маркетинг - Реклама - PR	<a href="#">×</a>
HR спеціалісти - Бізнес-тренери	<a href="#">×</a>
Наука - Освіта - Переклад	<a href="#">×</a>

Рисунок 3.31 – Заповнена форма додавання рубрики

Для додавання нової рубрики, адміністратор повинен натиснути кнопку «Зберегти».

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рубрика

Зберегти

Рубрики	Видалити
ІТ	<a href="#">×</a>
Маркетинг - Реклама - PR	<a href="#">×</a>
HR спеціалісти - Бізнес-тренери	<a href="#">×</a>
Наука - Освіта - Переклад	<a href="#">×</a>
Дизайн - Графіка - Фото	<a href="#">×</a>

Рисунок 3.32 – Результат додавання нової рубрики

Щоб видалити рубрику необхідно натиснути на червоний хрестик біля рубрики, яку адміністратор хоче видалити.

Ви зашли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рубрика

Зберегти

Рубрики	Видалити
ІТ	<a href="#">×</a>
Маркетинг - Реклама - PR	<a href="#">×</a>
HR спеціалісти - Бізнес-тренери	<a href="#">×</a>
Наука - Освіта - Переклад	<a href="#">×</a>

Рисунок 3.33 – Результат видалення рубрики

Щоб змінити інформацію про сфери діяльності, потрібно перейти на сторінку «Сфери діяльності».



Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Сфера діяльності	Рубрика	Видалити
IT - розробка програмного забезпечення	IT	X
HR	HR спеціалісти - Бізнес-тренери	X
Реклама та PR-послуги	Маркетинг - Реклама - PR	X

Рисунок 3.34 – Сторінка інформації про сфери діяльності

Щоб, додати нову сферу діяльності, адміністратор повинен вписати сферу діяльності в поле вводу і вибрати з списку рубрику, якій належить нова сфера діяльності.

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Сфера діяльності	Рубрика	Видалити
IT - розробка програмного забезпечення	IT	X
HR	HR спеціалісти - Бізнес-тренери	X
Реклама та PR-послуги	Маркетинг - Реклама - PR	X

Рисунок 3.35 – Заповнена форма додавання сфери діяльності

Для додавання нової сфери діяльності, адміністратор повинен натиснути кнопку «Зберегти».

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Сфера діяльності	Рубрика	Видалити
IT - розробка програмного забезпечення	IT	X
HR	HR спеціалісти - Бізнес-тренери	X
Реклама та PR-послуги	Маркетинг - Реклама - PR	X
Освіта	Наука - Освіта - Переклад	X

Рисунок 3.36 – Результат додавання нової сфери діяльності

Щоб видалити сферу діяльності необхідно натиснути на червоний хрестик біля назви сфери діяльності, яку адміністратор хоче видалити.

Ви зашли як **admin**
[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)




Сфера діяльності	Рубрика	Видалити
IT - розробка програмного забезпечення	IT	X
HR	HR спеціалісти - Бізнес-тренери	X
Реклама та PR-послуги	Маркетинг - Реклама - PR	X

Рисунок 3.37 – Результат видалення сфери діяльності

Щоб змінити інформацію про навички, потрібно перейти на сторінку «Навички».

Ви зашли як **admin**
[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)




Навик	Рубрика	Видалити
php	IT	X
html	IT	X
css	IT	X
JavaScript	IT	X
java	IT	X
python	IT	X
C#	IT	X

Рисунок 3.38 – Сторінка інформації про навички

Щоб, додати нову навичку, адміністратор повинен вписати назву навички у поле вводу та обрати рубрику, якій відповідає нова навичка.

Ви зашли як **admin**
[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)




Навик	Рубрика	Видалити
php	IT	X
html	IT	X
css	IT	X
JavaScript	IT	X
java	IT	X
python	IT	X
C#	IT	X

Рисунок 3.39 – Заповнена форма додавання міста

Для додавання нової навички, адміністратор повинен натиснути кнопку «Зберегти».

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Навик	Рубрика	Видалити
php	ІТ	X
html	ІТ	X
css	ІТ	X
JavaScript	ІТ	X
java	ІТ	X
python	ІТ	X
C#	ІТ	X
C++	ІТ	X

Рисунок 3.40 – Результат додавання нової навички

Щоб видалити навик необхідно натиснути на червоний хрестик біля навички, яку адміністратор хоче видалити.

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Навик	Рубрика	Видалити
php	ІТ	X
html	ІТ	X
css	ІТ	X
JavaScript	ІТ	X
java	ІТ	X
python	ІТ	X
C#	ІТ	X

Рисунок 3.41 – Результат видалення навички

Щоб змінити інформацію про університети, потрібно перейти на сторінку «Університети».

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

Рівень мови	Видалити
КПІ ім. Ігоря Сікорського	X

Рисунок 3.42 – Сторінка інформації про університети

Щоб, додати новий університет, адміністратор повинен вписати назву міста університету в поле для вводу.

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

**Рівень мови**
**Видалити**

КПІ ім. Ігоря Сікорського X

Рисунок 3.43 – Заповнена форма додавання університету

Для додавання нового університету, адміністратор повинен натиснути кнопку «Зберегти».

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

**Рівень мови**
**Видалити**

КПІ ім. Ігоря Сікорського X

Національний авіаційний університет X

Рисунок 3.44 – Результат додавання нового університету

Щоб видалити університет необхідно натиснути на червоний хрестик біля назви університету, який адміністратор хоче видалити.

Ви зайшли як **admin**

[Вихід](#)
[Головна](#)
[Міста](#)
[Курси](#)
[Рівні освіти](#)
[Типи зайнятості](#)
[Рівень навичок](#)
[Мови](#)
[Рівень мови](#)
[Рубрики](#)
[Сфери діяльності](#)
[Навички](#)
[Університети](#)

**Рівень мови**
**Видалити**

КПІ ім. Ігоря Сікорського X

Рисунок 3.45 – Результат видалення університету

#### 4 РЕКОМЕНДАЦІЇ ПО ОСВОЄННЮ

Для кращого розуміння роботи із застосунком, користувач повинен мати навички роботи з веб-сервісами, а також бути ознайомленим з даною інструкцією.

					КПІ.ІП-5114.045440.06.34	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дат		

**ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ**

**Кафедра автоматизованих систем обробки інформації і управління**

«ЗАТВЕРДЖЕНО»

В.о. завідувача кафедри

\_\_\_\_\_ Павлов О.А.  
(підпис) (ініціали, прізвище)

“ ” \_\_\_\_\_ 2019 р.

**Web-застосування для пошуку роботи**

**Осиграни**

КП.ІП5114.045440.02.13

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ О.М. Клименко

Нормоконтроль:

\_\_\_\_\_ К.І. Ліщук

Виконавці:

\_\_\_\_\_ М.О. Миколаєнко

Київ – 2019 року

*Тексти програмного коду*

**Web-застосування для роботи**

(Найменування програми (документа))

*CD-R*

(Вид носія даних)

*20 арк, 35800 кб.*

(Обсяг програми (документа) , арк.,) Кб)

Київ - 2019

					КПІ.ІП-5114.045440.02.13	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дат		

**Опис програми**

// Database.php

Class Database{

public \$host = DB\_HOST;

public \$user = DB\_USER;

public \$pass = DB\_PASS;

public \$dbname = DB\_NAME;

public \$link;

public \$error;

public function \_\_construct(){

\$this-&gt;connectDB();

}

//з'єднання з базою даних

private function connectDB(){

\$this-&gt;link = new mysqli(\$this-&gt;host, \$this-&gt;user, \$this-&gt;pass,

\$this-&gt;dbname);

if(!\$this-&gt;link){

\$this-&gt;error = "Connection fail". \$this-&gt;link-&gt;connect\_error;

return false;

}

}

// вибірка даних з бази даних

public function select(\$query){

\$result = \$this-&gt;link-&gt;query(\$query) or die(\$this-&gt;link-&gt;error.\_\_LINE\_\_);

if(\$result-&gt;num\_rows &gt; 0){

return \$result;

} else {

return false;

}

					КПІ.ІП-5114.045440.02.13	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дат		



```

}

// вставка даних в базу даних
public function insert($query){
    $insert_row = $this->link->query($query) or die($this->link->error.__LINE__);
    if($insert_row){
        return $insert_row;
    } else {
        return false;
    }
}

// оновлення даних в базі даних
public function update($query){
    $update_row = $this->link->query($query) or die($this->link->error.__LINE__);
    if($update_row){
        return $update_row;
    } else {
        return false;
    }
}

// видалення даних з бази даних
public function delete($query){
    $delete_row = $this->link->query($query) or die($this->link->error.__LINE__);
    if($delete_row){
        return $delete_row;
    } else {
        return false;
    }
}

```

```
// Format.php
class Format
{
    //вибірка року з дати
    public function formatGetYear($date){
        return date('Y', strtotime($date));
    }

    //форматування дати для запису в базу даних
    public function formatDateToDB($date){
        return date('Y-m-d', strtotime($date));
    }

    //форматування дати в нормальний вигляд
    public function formatDateNormal($date){
        return date('d.m.Y', strtotime($date));
    }

    //вибірка місяця і року з дати
    public function formatMonthYearNormal($date){
        return date('m.Y', strtotime($date));
    }

    //вибірка дня з дати
    public function formatGetDay($date){
        return ltrim(date('d',strtotime($date)), '0');
    }

    //вибірка місяця з дати
    public function formatGetMonth($date){
        return ltrim(date('m',strtotime($date)), '0');
    }
}
```

```

        //валідація даних
        public function validation($data){
            $data = trim($data);
            $data = stripslashes($data);
            $data = htmlspecialchars($data);
            return $data;
        }
    }
}

```

```

// Session.php

```

```

class Session{

    //ініціалізація сесії
    public static function init(){
        session_start();
    }

    //присвоєння сесії значення
    public static function set($key, $val){
        $_SESSION[$key] = $val;
    }

    //отримати значення сесії
    public static function get($key){
        if (isset($_SESSION[$key])) {
            return $_SESSION[$key];
        } else {
            return false;
        }
    }
}

```

					КПІ.ІП-5114.045440.02.13	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дат		

```
//перевірка сесії користувача
public static function checkSession(){
    self::init();
    if (self::get("user")== false) {
        self::destroy();
        header("Location:index.php");
    }
}

//перевірка сесії адміністратора
public static function checkAdminSession(){
    self::init();
    if (self::get("admin")== false) {
        self::destroy();
        header("Location:index.php");
    }
}

//перевірка авторизації
public static function checkLogin(){
    self::init();
    if (self::get("user")== true) {
        header("Location:Cabinet.php");
    }
}

//знищення сесії
public static function destroy(){
    session_destroy();
    session_unset();
}
}
```

```
// System.php
require_once 'Session.php';
require_once 'Database.php';
require_once 'Format.php';

class System
{
    private $db;
    private $fm;

    public function __construct()
    {
        $this->db = new Database();
        $this->fm = new Format();
    }

    //перевірка типу користувача за ідентифікатором
    public function checkTypeUser($id){
        $query = "SELECT `type` FROM `users` WHERE `id` = $id LIMIT 1";
        $user_type = $this->db->select($query);
        if($user_type){
            $result = $user_type->fetch_assoc();
            return $result['type'];
        }
    }

    //вибірка списку міст
    public function getCities(){
        $query = "SELECT `id`, `city`, `city_en` FROM `cities`";
        $cities = $this->db->select($query);
        return $cities;
    }

    //вибірка міста за ідентифікатором
```

					КПІ.ІП-5114.045440.02.13	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дат		

```

public function getCityById($id){
    $query = " SELECT `city` FROM `cities` WHERE `id` = '$id' LIMIT 1";
    $result = $this->db->select($query);
    $city = $result->fetch_assoc();
    return $city['city'];
}

```

//додавання міста

```

public function addCity($data){
    $city = $this->fm->validation($data['city']);
    $city = mysqli_real_escape_string($this->db->link, $city);
    $city_en = $this->fm->validation($data['city_en']);
    $city_en = mysqli_real_escape_string($this->db->link, $city_en);
    $query = "
        INSERT INTO `cities`
        SET `city` = '$city',
            `city_en` = '$city_en'
    ";
    $insert_row = $this->db->insert($query);
}

```

//видалення міста

```

public function deleteCity($id){
    $query = "DELETE FROM `cities` WHERE `id` = '$id' LIMIT 1";
    $del_citiy = $this->db->delete($query);
}

```

//вибірка курсу по ідентифікатору

```

public function getCourseById($id){
    $query = "SELECT `name`, `link` FROM `courses` WHERE `id` = '$id'";
    $select_row = $this->db->select($query);
    if($select_row){
        $result = $select_row->fetch_assoc();
        return $result;
    }
}

```

					КПІ.ІП-5114.045440.02.13	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

```

    }
}

//додавання курсу
public function addCourse($data){
    $name = $this->fm->validation($data['name']);
    $name = mysqli_real_escape_string($this->db->link, $name);
    $link = $this->fm->validation($data['link']);
    $link = mysqli_real_escape_string($this->db->link, $link);
    $skills = $data['skills'] ?? '';
    $query = "
        INSERT INTO `courses`
        SET `name` = '$name',
            `link` = '$link'
    ";
    $insert_row = $this->db->insert($query);
    if($insert_row){
        $course_id = $this->db->link->insert_id;
        foreach ($skills as $skill_id) {
            $query2 = "
                INSERT INTO `course_skills`
                SET `course_id` = '$course_id',
                    `skill_id` = '$skill_id'
            ";
            $insert_row2 = $this->db->insert($query2);
        }
    }
}

//видалення курсу
public function deleteCourse($id){
    $course_id = $this->getCourseId($id);
    $query = "DELETE FROM `course_skills` WHERE `id`=' $id' LIMIT 1";
    $del_course_skills = $this->db->delete($query);
}

```

					КПІ.ІП-5114.045440.02.13	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дат		

```

$query = "DELETE FROM `courses` WHERE `id`='$course_id' LIMIT 1";
$del_courses = $this->db->delete($query);
}

//отримання ідентифікатора курсу
public function getCourseId($id){
    $query = "SELECT `course_id` FROM `course_skills` WHERE `id` = '$id'";
    $select_row = $this->db->select($query);
    if($select_row){
        $result = $select_row->fetch_assoc();
        return $result['course_id'];
    }
}

//отримання списку навичок
public function getSkills(){
    $query = "SELECT `id`, `name`, `rubric_id` FROM `skills`";
    $skills = $this->db->select($query);
    return $skills;
}

//вибірка навичок за ідентифікатором
public function getSkillsById($id){
    $query = " SELECT `name` FROM `skills` WHERE `id` IN ($id)";
    $select_row = $this->db->select($query);
    $Arr_skills = array();
    if($select_row){
        while($result = $select_row->fetch_assoc()){
            array_push($Arr_skills, $result['name']);
        }
    }
    return $Arr_skills;
}

```



//додавання навичок

```

public function addSkills($data){
    $name = $this->fm->validation($data['name']);
    $name = mysqli_real_escape_string($this->db->link, $name);
    $rubric_id = $data['rubric'] ?? "";
    $query = "
        INSERT INTO `skills`
        SET `name`          = '$name',
            `rubric_id`     = '$rubric_id'
    ";
    $insert_row = $this->db->insert($query);
}

```

//видалення навичок

```

public function deleteSkills($id){
    $query = "DELETE FROM `skills` WHERE `id`='$id' LIMIT 1";
    $del_skills = $this->db->delete($query);
}

```

//вибірка рівнів навичок

```

public function getExperienceSkills(){
    $query = "SELECT `id`, `name` FROM `experience_skill`";
    $experience_skills = $this->db->select($query);
    return $experience_skills;
}

```

//вибірка курсів і навичок

```

public function getCoursesSkills(){
    $query = "SELECT * FROM `course_skills`";
    $result = $this->db->select($query);
    $Arr_courses = array();
    if($result){
        while($courses = $result->fetch_assoc()){

```

```

        $Arr_courses[] = array('id'=> $courses['id'], 'course'=>$this->getCourseById($courses['course_id']), 'skills'=>$this->getSkillsById($courses['skill_id']));
    }
}
return $Arr_courses;
}

```

//вибірка рівнів освіти

```

public function getEducationLevel(){
    $query = "SELECT `id`, `level` FROM `education_level`";
    $education_level = $this->db->select($query);
    return $education_level;
}

```

//додавання рівня освіти

```

public function addEducationLevel($data){
    $level = $this->fm->validation($data['level']);
    $level = mysqli_real_escape_string($this->db->link, $level);
    $query = "
        INSERT INTO `education_level`
        SET `level` = '$level'
    ";
    $insert_row = $this->db->insert($query);
}

```

//видалення рівня освіти

```

public function deleteEducationLevel($id){
    $query = "DELETE FROM `education_level` WHERE `id`=' $id' LIMIT 1";
    $del_education_level = $this->db->delete($query);
}

```

//вибірка типів зайнятості

```

public function getEmployments(){
    $query = "SELECT `id`, `employment` FROM `employments`";
}

```

					КПІ.ІП-5114.045440.02.13	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дат		

```

        $employments = $this->db->select($query);
        return $employments;
    }

    //отримання типу зайнятості за ідентифікатором
    public function getEmploymentById($id){
        $query = " SELECT `employment` FROM `employments` WHERE `id` = '$id'
LIMIT 1";

        $result = $this->db->select($query);
        if($result){
            $employment = $result->fetch_assoc();
            return $employment['employment'];
        }
    }

    //додавання типу зайнятості
    public function addEmployment($data){
        $employment = $this->fm->validation($data['employment']);
        $employment = mysqli_real_escape_string($this->db->link, $employment);
        $query = "
            INSERT INTO `employments`
            SET `employment` = '$employment'
        ";
        $insert_row = $this->db->insert($query);
    }

    //видалення типу зайнятості
    public function deleteEmployment($id){
        $query = "DELETE FROM `employments` WHERE `id`=' $id' LIMIT 1";
        $del_employment = $this->db->delete($query);
    }

    //отримання назви навички за ідентифікатором
    public function getSkillName($id){

```

```

$query = " SELECT `name` FROM `skills` WHERE `id` = '$id' LIMIT 1";
$result = $this->db->select($query);
$skill = $result->fetch_assoc();
return $skill['name'];
}

//отримання назви рівня навички за ідентифікатором
public function getExperienceSkillName($id){
    if(!empty($id)){
        $query = " SELECT `name` FROM `experience_skill` WHERE `id` = '$id'
LIMIT 1";

        $result = $this->db->select($query);
        $experience_skill = $result->fetch_assoc();
        return $experience_skill['name'];
    }
}

//додавання рівня навички
public function addExperienceSkill($data){
    $name = $this->fm->validation($data['name']);
    $name = mysqli_real_escape_string($this->db->link, $name);
    $query = "
        INSERT INTO `experience_skill`
        SET `name` = '$name'
    ";
    $insert_row = $this->db->insert($query);
}

//видалення рівня навички
public function deleteExperienceSkill($id){
    $query = "DELETE FROM `experience_skill` WHERE `id`=' $id' LIMIT 1";
    $del_experience_skill= $this->db->delete($query);
}

```

//вибірка списку мов

```

public function getLanguage(){
    $query = "SELECT `id`, `language` FROM `language`";
    $language = $this->db->select($query);
    return $language;
}

```

//додавання мови

```

public function addLanguage($data){
    $language = $this->fm->validation($data['language']);
    $language = mysqli_real_escape_string($this->db->link, $language);
    $query = "
        INSERT INTO `language`
        SET `language` = '$language'
    ";
    $insert_row = $this->db->insert($query);
}

```

//видалення мови

```

public function deleteLanguage($id){
    $query = "DELETE FROM `language` WHERE `id`='$id' LIMIT 1";
    $del_language = $this->db->delete($query);
}

```

```

public function getLanguageLevel(){
    $query = "SELECT `id`, `level` FROM `language_level`";
    $language_level = $this->db->select($query);
    return $language_level;
}

```

//додавання рівня володіння мови

```

public function addLanguageLevel($data){
    $level = $this->fm->validation($data['level']);
    $level = mysqli_real_escape_string($this->db->link, $level);
}

```

					КПІ.ІП-5114.045440.02.13	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дат		

```

$query = "
    INSERT INTO `language_level`
    SET `level` = '$level'
";
$insert_row = $this->db->insert($query);
}

//видалення рівня володіння мови
public function deleteLanguageLevel($id){
    $query = "DELETE FROM `language_level` WHERE `id`='$id' LIMIT 1";
    $del_language_level = $this->db->delete($query);
}

//вибірка списку рубрик
public function getRubrics(){
    $query = "SELECT `id`, `name` FROM `rubrics`";
    $rubrics = $this->db->select($query);
    return $rubrics;
}

//отримання рубрики за ідентифікатором
public function getRubricById($id){
    $query = " SELECT `name` FROM `rubrics` WHERE `id` = '$id' LIMIT 1";
    $result = $this->db->select($query);
    if($result){
        $rubric = $result->fetch_assoc();
        return $rubric['name'];
    }
}

//додавання рубрики
public function addRubrics($data){
    $name = $this->fm->validation($data['name']);
    $name = mysqli_real_escape_string($this->db->link, $name);

```

					КПІ.ІП-5114.045440.02.13	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дат		

```

$query = "
    INSERT INTO `rubrics`
    SET `name` = '$name'
";
$insert_row = $this->db->insert($query);
}

//видалення рубрики
public function deleteRubrics($id){
    $query = "DELETE FROM `rubrics` WHERE `id`='$id' LIMIT 1";
    $del_rubrics = $this->db->delete($query);
}

//отримання списку сфер діяльності
public function getScopes(){
    $query = "SELECT `id`, `name`, `rubric_id` FROM `scopes`";
    $scopes = $this->db->select($query);
    return $scopes;
}

//додавання сфери діяльності
public function addScopes($data){
    $name = $this->fm->validation($data['name']);
    $name = mysqli_real_escape_string($this->db->link, $name);
    $rubric_id = $data['rubric'] ?? "";
    $query = "
        INSERT INTO `scopes`
        SET `name` = '$name',
            `rubric_id` = '$rubric_id'
    ";
    $insert_row = $this->db->insert($query);
}

//видалення сфери діяльності

```

					КПІ.ІП-5114.045440.02.13	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дат		

```

public function deleteScopes($id){
    $query = "DELETE FROM `scopes` WHERE `id`='{$id}' LIMIT 1";
    $del_scopes = $this->db->delete($query);
}

```

//вибірка списку університетів

```

public function getUniversity(){
    $query = "SELECT `id`, `name` FROM `university`";
    $university = $this->db->select($query);
    return $university;
}

```

//додавання університету

```

public function addUniversity($data){
    $name = $this->fm->validation($data['name']);
    $name = mysqli_real_escape_string($this->db->link, $name);
    $query = "
        INSERT INTO `university`
        SET `name` = '$name'
    ";
    $insert_row = $this->db->insert($query);
}

```

//видалення університету

```

public function deleteUniversity($id){
    $query = "DELETE FROM `university` WHERE `id`='{$id}' LIMIT 1";
    $del_university = $this->db->delete($query);
}

```

//пошук курсів за ідентифікатором навички

```

public function findCoursesSkill($id){
    $query = "SELECT `course_id` FROM `course_skills` WHERE `skill_id` = '{$id}'";
    $course_skills = $this->db->select($query);
    return $course_skills;
}

```



```

    }

    //пошук курсів за навчиками
    public function findCoursesBySkill($id){
        $Arr_course_skills = array();
        $course_skills = $this->findCoursesSkill($id);
        if($course_skills){
            while($result = $course_skills->fetch_assoc()){
                $Arr_course_skills[] = $this->getCourseById($result['course_id']);
            }
        }
        return $Arr_course_skills;
    }

    //отримати статистику з таблиці
    public function getStatisticTable($table){
        $query = "SELECT `name`, `count` FROM $table";
        $statistic = $this->db->select($query);
        return $statistic;
    }

    //вибірка списку місяців
    public function getMonths(){
        $query = "SELECT `month`, `number_month` FROM `months`";
        $months = $this->db->select($query);
        return $months;
    }
}

```

**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматизованих систем обробки інформації і управління**

**“ЗАТВЕРДЖЕНО”**

**В.о. завідувача кафедри**

\_\_\_\_\_ **О.А. Павлов**

**“    ”    \_\_\_\_\_ 2019 р.**

**Web-застосування для пошуку роботи**

**Програма та методика тестування**

**КПІ.ІП-5114.045440.04.51**

**“ПОГОДЖЕНО”**

**Керівник проекту:**

\_\_\_\_\_ **О.М. Клименко**

**Нормоконтроль:**

\_\_\_\_\_ **К.І. Ліщук**

**Виконавець:**

\_\_\_\_\_ **М.О. Миколаєнко**

**Київ – 2019 року**

**ЗМІСТ**

<b>1</b>	<b>АНАЛІЗ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>3</b>
<b>2</b>	<b>ПІДХОДИ ДО ТЕСТУВАННЯ .....</b>	<b>5</b>
2.1	Інтеграційне тестування.....	5
2.2	Компонентне тестування.....	5
2.3	Тестування продуктивності .....	5
2.4	Тестування функціональності .....	5
2.5	Тестування зручності користування .....	6
2.6	Тестування безпеки .....	6
<b>3</b>	<b>КРИТЕРІЇ ПРОХОДЖЕННЯ ТЕСТУВАННЯ.....</b>	<b>7</b>
<b>4</b>	<b>ПРОЦЕС ТЕСТУВАННЯ.....</b>	<b>8</b>
4.1	Дані до функціональних тестів .....	8
4.2	Задачі тесту .....	8

Змн.	Арк.	№ докум.	Підпис	Дата

## 1 АНАЛІЗ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для того, щоб проаналізувати якість програмного забезпечення необхідно його протестувати. Тестування є одним з найголовніших етапів в розробці програмних продуктів. Тестування дозволяє завчасно виявити неправильну роботу програми, до того, як її почнуть масово використовувати, та вчасно виправити помилки при розробці. Адже на ранніх етапах легше виправити неточності в роботі програми, ніж коли вона вже використовується. Виправлення можуть зайняти декілька днів, і можуть вплинути на роботу інших модулів.

У даному плані будуть протестовані наступні функції:

- реєстрація користувача;
- авторизація користувача;
- створення резюме;
- редагування резюме;
- видалення резюме;
- створення PDF файлу резюме;
- завантаження PDF файлу резюме;
- перегляд резюме;
- створення вакансії;
- редагування вакансії;
- видалення вакансії;
- перегляд вакансій;
- робота з API dou.ua;
- робота з API jooble.org;
- перегляд списку курсів;
- перегляд статистики;
- додавання нових даних адміністратором у таблиці бази даних;

					КПІ.ІП-5114.045440.04.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

– видалення старих даних адміністратором з таблиць бази даних.

Тестові модулі:

- реєстрація користувача;
- реєстрація користувача з існуючим логіном;
- реєстрація користувача з некоректним паролем;
- авторизація користувача;
- авторизація користувача з неіснуючим логіном;
- авторизація користувача з некоректним паролем;
- створення резюме;
- створення резюме з незаповненими обов'язковими полями;
- редагування резюме;
- редагування резюме з пустими обов'язковими полями;
- видалення резюме;
- створення PDF файлу резюме;
- завантаження PDF файлу резюме;
- перегляд резюме;
- створення вакансії;
- створення вакансії з незаповненими обов'язковими полями;
- редагування вакансії;
- редагування вакансії з незаповненими обов'язковими полями;
- видалення вакансії;
- перегляд вакансій;
- отримання даних від API dou.ua;
- отримання даних від API jooble.org;
- перегляд списку курсів;
- перегляд статистики;
- додавання нових даних адміністратором у таблиці бази даних;
- видалення старих даних адміністратором з таблиць бази даних.

## 2 ПІДХОДИ ДО ТЕСТУВАННЯ

Даний план включає використання наступних методів:

- інтеграційного;
- компонентного;
- продуктивності;
- функціонального;
- тестування зручності користування;
- тестування безпеки.

### 2.1 Інтеграційне тестування

Методом інтеграційного тестування перевіряється робота клієнтської частини з серверною частиною веб-застосунку.

### 2.2 Компонентне тестування

Методом компонентного тестування буде перевірено логічні частини сервера:

- доступ до бази даних;
- робота з API.

### 2.3 Тестування продуктивності

Методом тестування продуктивності буде перевірена швидкодія роботи з API.

### 2.4 Тестування функціональності

Методом функціонального тестування буде перевірено відповідність функціональним вимогам:

- тестування введенням граничних та недопустимих значень в поля, які
- 

					КПІ.ІП-5114.045440.04.51	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

можна редагувати;

- тестування на відповідність функціональним вимогам.

## 2.5 Тестування зручності користування

Методом тестування зручності користування буде перевірено наскільки користувачу зручно працювати з веб-додатком:

- тестування зручності використання;
- тестування інтерфейсу.

## 2.6 Тестування безпеки

Методом тестування безпеки буде перевірено безпеку веб-додатку:

- тестування конфіденційності;
- тестування доступності.

### 3 КРИТЕРІЇ ПРОХОДЖЕННЯ ТЕСТУВАННЯ

Критерієм проходження інтеграційного, компонентного, функціонального методів тестування є повне успішне виконання кожного пункту тесту. Якщо хоч один тест було виконано не успішно тестування вважається проваленим.

Критерієм проходження тестування продуктивності є успішне виконання кожного пункту тесту за допустимий час. В разі перевищення допустимого на виконання часу або не відповіді на запит взагалі тест вважається проваленим.

Критерієм проходження тестування зручності користувача є зручність веб-додатку у використанні. В разі незручності використання веб-додатку тест вважається проваленим.

Критерієм проходження тестування безпеки є заборона несанкціонованого доступу до конфіденційних даних. В разі доступу користувача до чужих конфіденціальних даних, або до кабінету адміністратора тест вважається проваленим.

					КПІ.ІП-5114.045440.04.51	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7



## 4 ПРОЦЕС ТЕСТУВАННЯ

### 4.1 Дані до функціональних тестів

Вхідними даними інтеграційного тесту є запити від клієнтської частини до серверної частини веб-додатку. Вихідними даними є відповідь від сервера.

Вхідними даними компонентного тесту є набори параметрів та визначених результатів, які і є вихідними даними компонентного тесту.

Вхідними даними продуктивного тесту є набори різноманітних даних, які імітують функціонування системи. Вихідними даними являються дані про швидкість обробки запитів, дані по навантаженню, тощо.

Вхідними даними функціонального тесту є дані, які імітують звичайну роботу з системою. Вихідними даними є відповідна реакція системи на отримані дані.

### 4.2 Задачі тесту

Задачею будь-якого тесту є перевірка правильності роботи програми в умовах виконання тесту, а також виявлення потенційних помилок у роботі.

**ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ**  
**Кафедра автоматизованих систем обробки інформації і управління**

«ЗАТВЕРДЖЕНО»  
В.о. завідувача кафедри  
\_\_\_\_\_ О.А. Павлов  
(підпис) (ініціали, прізвище)  
“ ” \_\_\_\_\_ 2019 р.

**Web-застосування для пошуку роботи**

**Графічні матеріали**

КП.ІП-5114.045440.07.99

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ О.М. Клименко

Нормоконтроль:

\_\_\_\_\_ К.І. Ліщук

Виконавці:

\_\_\_\_\_ М.О. Миколаєнко

Київ – 2019 року